(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) **Offenlegungsschrift**

(11) **DE 3506118 A1**

(51) Int. Cl. 4:
G 06 F 15/16
G 06 F 11/28
G 06 F 13/38
B 60 R 16/02

(21) Aktenzeichen: P 35 06 118.9
(22) Anmeldetag: 22. 2. 85
(43) Offenlegungstag: 28. 8. 86

(71) Anmelder:
Robert Bosch GmbH, 7000 Stuttgart, DE

(72) Erfinder:
Botzenhardt, Wolfgang, Dipl.-Ing., 7320 Göppingen, DE; Dais, Siegfried, Dipl.-Phys., 7016 Gerlingen, DE; Kiencke, Uwe, Dipl.-Ing., 7140 Ludwigsburg, DE; Litschel, Martin, Dipl.-Ing., 7143 Vaihingen, DE; Krampe, Wolfgang, Dipl.-Ing., 7015 Korntal-Münchingen, DE

(54) Verfahren zum Betreiben einer Datenverarbeitungsanlage für Kraftfahrzeuge

Es wird ein Verfahren zum Betreiben einer Datenverarbeitungsanlage für Kraftfahrzeuge mit wenigstens zwei Rechnern und eine die Rechner verbindenden Leitung zum Übertragen von Botschaften vorgeschlagen, mit deren Hilfe eine schnelle und sichere Datenübertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeräte-Kopplung im Kraftfahrzeug möglich ist. Es ist ein Ausführungsbeispiel angegeben, das die Schnittstelle zwischen den einzelnen Rechnern und der die Rechner verbindenden Leitung ausführlich beschreibt und mit dessen Hilfe eine Steuergeräte-Kopplung im Kraftfahrzeug realisiert werden kann.

DE 3506118 A1

DE 3506118 A1

6. Ansprueche

1. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages), dadurch gekennzeichnet, dass sich die Botschaften bezueglich Inhalt und Bedeutung mittels eines Identifizieres selbst identifizieren.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass dem Identifizierer eine Prioritaet insbesondere fuer die Reihenfolge der Uebertragung der Botschaften auf der Leitung zugeordnet ist.

3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, dass der Identifizierer mit impliziter Prioritaetenangabe vorzugsweise am Anfang der Botschaft steht.

4. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der Identifizierer Inhalte wie Adressen, Daten, Sensorsignale, Stellgroessen, Zwischenergebnisse, Befehle fuer Synchronisation, Befehle zum Ausloesen von Aktionen kennzeichnet.

5. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der Identifizierer Inhalte wie Drehzahl, Drehzahlgradient, Motortemperatur, Last der Antriebsmaschine usw. kennzeichnet.

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass der Identifizierer Befehlsdaten , zum Beispiel von einer Antiblockiereinrichtung, einer Antischlupfeinrichtung usw. kennzeichnet.

7. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) und einer Fehlererkennung bezueglich der Botschaft, dadurch gekennzeichnet, dass die Botschaft mit einem definierten Bitmuster (wenigstens ein Bit) anfaengt, dieses Bitmuster mit in den CRC-Check mit einbezogen wird und dass auf den CRC-Check (Kontrollwort) ein Bitmuster folgt, das zum Anfangsbitmuster komplementaer ist.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass das zur Berechnung des Kontrollworts benuzte Generator-Polynom so gestaltet wird, dass vorzugsweise ein

BCH-Code (Bose, Chaudhure, Hocquenghem) mit dem Restklassen - Polynom (X + 1) multipliziert wird.

9. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) und einer Fehlermeldung (error report) ,dadurch gekennzeichnet, dass auf die Leitung (Bus) dominate und rezessive Zustaende uebertragen werden und dass die Fehlermeldung aus einer einzigen, mit der bei der Uebertragung der Botschaft auftretenden Zustandsfolge nicht zu verwechselnden Zustandsfolge dominanter Zustaende besteht.

10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die Fehlermeldung zu jedem beliebigen Zeitpunkt (zB. waehrend der laufenden Uebertragung einer Botschaft) gesendet werden kann.

11. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass jeder Teilnehmer, der mit der Leitung gekoppelt ist, eine Fehlermeldung abgeben kann.

12. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass das Ende der Fehlermeldung zur zeitlichen Synchronisation aller Teilnehmer dient.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, dass bei mehreren Fehlermeldungen das Ende der letzten der Synchronisation dient.

14. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass der informationstragende Teil der Botschaft mit einer nicht dominanten Zustandsfolge abgeschlossen wird, sodass im Fehlerfall die aus dominanten Zustaenden bestehende Fehlermeldung zeitlich in die Uebertragung der nicht dominanten Zustandsfolge trifft und dadurch eindeutig die fehlerhafte Botschaft bestimmt ist.

15. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die nicht dominante Zustandsfolge am Ende der Botschaft um mindestens einen Zustand laenger ist als die fuer die Fehlererkennung erforderliche minimale Zustandsfolge.

16. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer

die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) ,dadurch gekennzeichnet, dass jeder Teilnehmer alle Botschaften empfaengt und nur die fuer ihn relevanten Botschaften weiter verarbeitet, indem jeder Teilnehmer eine Liste der fuer ihn relevanten Botschaften fuehrt und Teile der empfangenen Botschaften mit den in der Liste enthaltenen Informationen in Beziehung bringt.

17. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass das in Beziehung bringen so erfolgt, dass alle Teilnehmer zum gleichen Rasterpunkt der Signaluebertragung die Entscheidung treffen koennen, ob die spezielle Botschaft weiter verfolgt wird.

18. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass die in den Botschaften enthaltenen Informationen in Abhaengigkeit von der in jedem Teilnehmer vorliegenden Liste in Speicherzellen uebertragbar sind und dem Teilnehmer zur Weiterverarbeitung zur Verfuegung stehen.

19. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass abhaengig vom Listeninhalt beim Empfang einer Botschaft eine Unterbrechungsanforderung ausloesbar ist.

20. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass die Uebernahme der Botschaft erst nach fehlerfreiem Empfang insbesondere durch alle Teilnehmer erfolgt.

21. Verfahren nach Anspruch 18, dadurch gekennzeichnet, dass die Informationsannahme blockierbar ist.

22. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass die Reihenfolge der Liste fuer die interne Weiterverarbeitung im Teilnehmer prioritaetsbestimmend ist.

23. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass die Weitergabe von Informationen, insbesondere die Abspeicherung, so erfolgt, dass inhaltlich zusammenhaengende Informationsteile im zugehoerigen Informationsverarbeitungsvorgang zusammengehoerig verarbeitet werden koennen (Konsistenz der Informationsbehandlung).

24. Verfahren nach Anspruch 16, dadurch gekennzeichnet, dass in jedem Zeitpunkt die Rangfolge der Liste die Prioritaet der Weiterverarbeitung bestimmt.

25. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) ,dadurch gekennzeichnet, dass jeder Teilnehmer den zu uebertragenden Botschaften mittels einer Liste eine Prioritaet zuordnet und in jedem Zeitpunkt die Liste die Prioritaet der im Teilnehmer zur Uebertragung bereitstehenden Botschaften bestimmt.

26. Verfahren nach Anspruch 25, dadurch gekennzeichnet, dass der Liste weitere Informationen insbesondere bezueglich des Uebertragungszustandes zugeordnet sind.

27. Verfahren nach Anspruch 25, dadurch gekennzeichnet, dass fehlerbehaftete Uebertragungen erfasst und registriert werden.

28. Verfahren nach Anspruch 27, dadurch gekennzeichnet, dass die Uebertragung bei auftretendem Fehler wiederholt wird.

29. Verfahren nach Anspruch 28, dadurch gekennzeichnet, dass nach einer vorbestimmten Anzahl von fehlerhaften Uebertragungen die Wiederholung der Uebertragung abgebrochen wird und eine Unterbrechungsanforderung ausgeloest werden kann.

30. Verfahren nach Anspruch 25, dadurch gekennzeichnet, dass die Liste eine Information bezueglich Uebertragungsanforderungen enthaelt, die vom Teilnehmer selbst und auch von den uebrigen mit der Leitung gekoppelten Einrichtungen bestimmt werden kann.

31. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) ,dadurch gekennzeichnet, dass jedem Teilnehmer ein bestimmter Teil der Uebertragungskapazitaet der Leitung zugeordnet wird.

32. Verfahren zum Betreiben einer Datenverarbeitungsanlage fuer Kraftfahrzeuge mit wenigstens zwei Rechnern, einer die Rechner verbindenden Leitung zum Uebertragen von Botschaften (Messages) ,dadurch gekennzeichnet, dass nach dem Auftreten einer bestimmten Anzahl von Fehlermeldungen eines bestimmten Teilnehmers sich dieser Teilnehmer sendesignalmaesig von der Leitung abkoppelt.

33. Verfahren zur Uebertragung von zweiwertigen Informationen

(0 und 1) im Kraftfahrzeug, dadurch gekennzeichnet, dass die von einem konstanten Signalpegel (0) abweichende Information (1) mit alternierenden Signalpegeln (+,-) uebertragbar ist.

34. Verfahren nach Anspruch 33, dadurch gekennzeichnet, dass die alternierenden Signalpegel zur Fehlererkennung ueberwachbar sind.

3506118

1. Stand der Technik

In den letzten Jahren wurde die Funktion des Kraftfahrzeugs durch elektronische Steuerungen wesentlich verbessert. Durch eine digitale Motorelektronik konnte z.B. der Kraftstoffverbrauch reduziert und die Emmission von Schadstoffen verringert werden. Mit dem Anti- Blockier-System lassen sich die Bremswege bei Vollbremsung verkuerzen, wobei gleichzeitig die Lenkbarkeit des Fahrzeugs erhalten bleibt.

Weitere Funktionsverbesserungen sind fuer das Kraftfahrzeug in Zukunft insbesondere dadurch zu erreichen, dass die Steuerfunktionen nicht mehr einzeln fuer sich allein arbeiten, sondern untereinander vermascht werden. Die Schaltvorgaenge eines elektronisch gesteuerten Automatik- Getriebes lassen sich z.B. mit geringerem Verschleiss der Kupplungsbelaege und ohne spuerbaren Ruck durchfuehren, wenn im Schaltaugenblick das Motormoment durch einen entsprechenden Eingriff in die elektronische Motorsteuerung kurzzeitig reduziert wird.

Dazu ist es erforderlich, dass der Rechner fuer die elektronische Getriebesteuerung genau im richtigen Zeitpunkt entsprechende Daten an den Rechner fuer die elektronische Motorsteuerung uebertraegt.Bislang wurde dies mit Hilfe einer Reihe von einzelnen Signalleitungen erreicht.

Die Anzahl derartiger Signalleitungen wird jedoch bei umfangreicheren Systemen zu gross. Man benoetigt deshalb in diesem Fall eine schnelle Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern, die wenig Anschluess im Steuergeraete - Stecker benoetigt und bei der die Information in codierter Form uebertragen wird.

Zu diesem Zwecke wurden in der Vergangenheit lokale Netzwerke zur Kopplung von Mikroprozessoren, Minirechnern und Peripheriegeraeten in Steuerungen, insbesondere jedoch in nachrichtentechnischen Anwendungen entwickelt. So gibt es als Stand der Technik eine grosse Zahl verschiedener Uebertragungsprotokolle fuer die Kopplung von Mikrorechnern wie z.B. DDB [1], IIC [2], MUART [3], CSMA [4], SDLC [4] und HDLC [4].

[1] VALVO, DDB specification

[2] VALVO, Technische Informationen fuer die Industrie 811215

[3] INTEL, Microprocessor and Peripheral Handbook, 1983

[4] A. Tannenbaum, Computer Networks, Prentice/Hall International, 1981

2. Nachteile des Stands der Technik

In den oben genannten Protokollen sind die Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug nur ungenuegend beruecksichtigt.

2.1 Waehrend in der Nachrichten- und Rechnertechnik insbesondere groessere Datenpakete uebertragen werden, sind die typischen Laengen der Datenpakete im Kraftfahrzeug klein. Im Auto werden zwischen Steuergeraeten, Sensoren und Stellern vorzugsweise Messwerte, Zwischenergebnisse von Rechenalgorithmen und Signale zur zeitlichen Synchronisation ausgetauscht. Die Entwicklung eines fuer diese Anwendungen geeigneten Uebertragungsprotokolls fuehrt zwangslaeufig zu anderen Ergebnissen.

2.2 Die Steuerungen im Auto arbeiten unter Echtzeitbedingungen, d.h. die Rechenoperationen und Steuereingriffe muessen in bestimmten Zeitfenstern erfolgen, schritthaltend mit den Prozessen. Fuer das lokale Netzwerk ergibt sich daraus die Forderung, dass die Uebertragungsleitung nach einer kurzen Latenzzeit (typisch 200 Mikrosekunden) fuer wichtige Botschaften frei sein muss, um zulange Verzoegerungen zu vermeiden.

Beim genormten Ethernet - Protokoll dauert demgegenueber allein die Uebertragung einer einzigen Botschaft mindestens 580 Mikrosekunden, trotz der hohen Uebertragungsrate von 10 MHz. Erst nach dieser Zeit ist der Bus zur Uebertragung weiterer Botschaften frei. Beginnen dann mehrere Busteilnehmer gleichzeitig mit der Uebertragung, so kommt es zu einer Kollision mehrerer Botschaften auf dem Netzwerk. Der Zugriffskonflikt wird geloest, indem sich zunaechst alle Sender zurueckziehen und erst nach einer statistischen Wartezeit erneut mit der Uebertragung beginnen. Durch diese Massnahmen sind jedoch auch wiederholte Buszugriffskonflikte nicht auszuschliessen. Dadurch wird das zuverlaessige Einhalten von harten Zeitbedingungen, wie bei Kraftfahrzeug - Steuerungen erforderlich, unmoeglich.

2.3 Kraftfahrzeuge werden je nach Wunsch des Kaeufers mit einem unterschiedlichen Ausstattungsumfang ausgeruestet. Die Konfiguration eines lokalen Netzwerkes und die Zahl seiner Teilnehmer muss deshalb auf einfache Weise aenderbar sein. Insbesondere ist es wichtig, dass die bereits am Netzwerk angeschlossenen Rechner mit unveraendertem Programm arbeiten koennen, wenn sich an der von ihnen realisierten Funktion nichts aendert. Das Uebertragungssystem muss so konzipiert sein, dass ueber die bestehenden, funktionalen Verknuepfungen zwischen

Steuergeraeten hinaus keine weiteren, durch das Busprotokoll bedingten Abhaengigkeiten eingefuehrt werden muessen. Diese Forderung ist bei keinem der bekannten Schnittstellen - Bausteinen erfuellt.

Z.B. werden bei DDB in jeder Botschaft die Sender- und Empfaenger- Adressen angegeben. Kommt ein weiterer Netzwerk - Teinehmer hinzu, der auch bereits auf dem Bus uebertragene Daten benoetigt, so muss in entsprechenden, bereits vorhandenen Teilnehmern die neue Adresse ergaenzt werden. Zusaetzlich muessen die bereits zu anderen Teilnehmern uebertragenen Daten nochmals auch zu den neuen gesendet werden. Obwoh von der logischen Struktur nicht erforderlich, erhaelt man eine grosse Vielzahl von Programmvarianten.

2.4 Viele bekannte Netzwerke (z.B. auf Basis Intel Mikroprozessor 8044, HDLC/SDLC - Protokoll) arbeiten nach dem sogenannten Master / Slave - Prinzip, dh. nur einer der Teinehmer (der Master) hat zu einem Zeitpunkt die Berechtigung zum Buszugriff. Damit umgeht man die sonst erforderliche Arbitrierung.

Im allgemeinen wird die Master - Eigenschaft nacheinander an alle Bus - Teilnehmer weitergegeben. Bei Automobil - Anwendungen ist ein derartiges Verfahren aber nachteilig. Ein Teilnehmer kann naemlich nur dann senden, wenn er gerade im Besitz der Sendeberechtigung ist. Dadurch koennen ggf. nicht tolerierbar lange Wartezeiten in den Slaves entstehen, bis eine Botschaft hoher Prioritaet uebertragen wird.

2.5 Unguenstig ist das Master / Slave - Konzept auch bei elektromagnetischen Stoerungen, mit denen im Kraftfahrzeug bekanntlich gerechnet werden muss. Dabei kann z.B. die Sendeberechtigung durch eine Stoerung verloren gehen oder irrtuemlicherweise ein weiterer Teilnehmer gleichzeitig eine Sendeberechtigung erlangen.

Die Bewaeltigung solcher Stoerfaelle ist zwar im Prinzip moeglich, erfordert aber viel Zeit (Busausfallzeit,CPU - Rechenzeit) und Aufwand (Hardware/Sofware), was im Hinblick auf die Echtzeit - Anforderungen der Steuergeraete nicht zugebilligt werden kann.

2.6 Ein Problem bei vielen bekannten Netzwerken ist die Synchronisation von Ereignissen. Sollen z.B. durch Uebertragungh einer Botschaft in zwei oder mehreren Teilnehmern gleichzeitig Aktionen ausgeloest werden, so muss die Botschaft von allen im genau gleichen Augenblick empfangen werden. Bei der sonst ueblichen, zeitlich aufeinanderfolgenden Uebertragung von Botschaften zu den einzelnen Empfaengern (Punkt zu Punkt - Verbindung) ist

die Gleichzeitigkeit des Empfangs einer gueltigen Botschaft prinzipiell nicht zu erreichen.

2.7 In Kfz - Netzwerken sind haeufig gleiche Botschaften an verschiedene Empfaenger zu senden. Der gleichzeitige Empfang durch alle angesprochenen Teilnehmer wuerde in diesen Faellen die Belegung des Netzwerkes betraechtlich reduzieren. Dieses Vorgehen wird aber von den bekannten Schnittstellen - Bausteinen nicht unterstuetzt.

2.8 Netzwerke im Kraftfahrzeug arbeiten in einer ausserordentlich stark gestoerten Umgebung. Zur Reduzierung der unerkannten Uebertragungsfehler ist deshalb eine leistungsfaehige Fehlererkennung erforderlich, so dass im Bedarfsfall eine Uebertragung wiederholt werden kann. Einfache Algorithmen wie die zusaetzliche Uebertragung von Quersummen - Bits erkennen nur Einzelfehler und sind nicht ausreichend bei den im Auto vorherschenden Bueschelstoerungen.

Selbst die aufwendigeren der heute bekannten Protokolle sind nicht in der Lage, ohne zusaetzliche Absicherungsprogramme auf Benutzerebene (z.B. Mehrfach -Uebertragung) eine ausreichende Uebertragungssicherheit im Kraftfahrzeug zu gewaehrleisten. So kann beim HDLC - Protokoll bereits ein Einzelbit - Fehler zu einer nicht erkennbaren Verfaelschung der Botschaft fuehren, obwohl diese durch einen CRC-Check von 16 Bit Laenge abgesichert wird.

2.9 Um eine Wiederholung im Fehlerfall veranlassen zu koennen, muss der Sender von allen Empfaengern eine Rueckmeldung ueber den korrekten Empfang der Botschaft erhalten. In den Punkten 2.6 und 2.7 wurde dargelegt, dass es von erheblichem Vorteil ist, die Botschaften gleichzeitig von mehreren Teinehmern empfangen zu koennen. Die Rueckmeldungen duerfen nun aber nicht wie bei bekannten Protokollen in einzelnen, aufeinaderfolgenden Botschaften oder Teilbotschaften erfolgen. Eine konsistente, gleichzeitige Behandlung der Daten durch die angesprochenen Bus - Teilnehmer ist in diesem Falle nicht zu gewaehrleisten.

2.10 Stoerungen in Kfz werden haeufig nur lokal wirksam. Aufgrund der endlichen Ausbreitungsgeschwindigkeit elektromagnetischer Wellen kann es z.B. vorkommen, dass nur ein Teil der Empfaenger in einem bestimmten Zeitintervall verfaelschte Pegel auf dem Bus abtastet. Bei gleichzeitigem Empfang einer Botschaft durch mehrere Teilnehmer koennte es also vorkommen, dass ein Teil der Empfaenger gueltige, der andere Teil verfaelschte Botschaften empfaengt. Dies darf nicht dazu fuehren, dass

in einem solchen Fehlerfall bereits einige Teilnehmer die Botschaft bearbeiten und damit ein synchrones Vorgehen aller Bus - Teilnehmer nicht mehr zustande kommt.

2.11 Die am lokalen Netzwerk angeschlossenen Teilnehmer muessen entscheiden, ob sie eine gerade uebertragene Botschaft empfangen wollen oder nicht. Aufgrund der starken Auslastung der Rechner durch Echtzeit - Aufgaben muss diese Entscheidung unbedingt per Hardware erfolgen. Bisher bekannte Schnittstellen - Bausteine fuer Mikrocomputer treffen diese Auswahlentscheidung nur an Hand der eigenen Teilnehmeradresse. Neben den bereits in Punkt 2.6 und 2.7 aufgefuehrten Nachteilen eines sochen Vorgehens wird der Rechner darueberhinaus stark mit dem Einordnen und inhaltsabhaengigen Abspeichern der Botschaften belastet. Die Rechner werden erst dann wirksam entlastet, wenn der Schnittstellen - Baustein empfangene Daten gleich inhaltsabhaengig in den zugeordneten Speicherzellen ablegt.

2.12 Zur Kodierung der im Kfz zu uebertragenden Daten wird abhaengig von der Anwendung eine unterschiedliche Anzahl von Bits verwendet. Unabhaengig von der Laenge der Daten muss jedoch sichergestellt werden, dass diese vom Schnittstellen - Baustein und vom angeschlossenen Rechner immer konsistent bearbeitet werden. Erfolgt z.B. die inhaltsbezogene Abspeicherung der Daten in mehreren Schritten, je nach Laenge der Daten, so ist sicherzustellen, dass bei zeitlicher Ueberlagerung von Rechnerzugriff und Uebertragung nicht alte und neue Daten derselben Bedeutung (Kennung) unzulaessigerweise vermischt werden.

Es werde z.B. eine Drehzahl in zwei Bytes codiert uebertragen und vom Schnittstellen - Baustein in zwei Speicherzellen abgelegt. Es darf dabei nicht passieren, dass der Rechner unerkannt das erste Byte der letzten und das zweite Byte der aktuellen Uebertragung ausliest und zu einer ungueltigen Drehzahl - Information zusammensetzt.

2.13 An ein lokales Netzwerk sind viele Sende- und Empfangsschaltungen angeschlossen. Die Ausfallwahrscheinlichkeit des Netzwerkes steigt bekanntlich mit der Anzahl dieser Schaltungen. Es muss deshalb verhindert werden, dass bereits der Ausfall eines einzelnen Teilnehmers das gesamte Netzwerk blockiert.

2.14 Die galvanische Kopplung mehrerer, raeumlich verteilter Steuergeraete fuehrt haeufig zu einer Beeintraechtigung deren Funktion, z.B. durch Ausgleichsstroeme.

Das Uebertragungsprotokoll fuer ein lokales Netzwerk im

Kfz muss so gestaltet sein, dass nicht nur mit teuren Lichtleitern, sondern auch bei Verwendung von Kupferleitern eine galvanische Entkopplung realisiert werden kann.

2.15 In Kfz - Anwendungen muessen sowohl schnell als auch langsam sich veraendernde Daten uebertragen werden. Um zu verhindern, dass der Informationsgehalt der schnell veraenderlichen Daten nicht mehr aktuell ist, wenn sie empfangen werden, muessen diese bevorzugt vor den anderen uebertragen werden. Deshalb muessen Buszugriffe, die Bearbeitung der Uebertragungsanforderungen im Sender und die Abarbeitung der empfangenen Daten nach Prioritaeten geordnet erfolgen. Die Prioritaeten sind den einzelnen Botschaften individuell zuzuordnen.

Ist z.B. die Anforderung zur Uebertragung der Motortemperatur bereits erfolgt, die Uebertragung selbst aber noch nicht geschehen, so muss dennoch die vorrangige Bearbeitung des nachtraeglich zur Uebertragung angemeldeten, hoeher prioren Lastsignals sichergestellt werden. Bei bekannten Schnittstellen kann die Bearbeitungsfolge nur vom Rechner per Programm festgelegt werden. Diese Vorgehensweise ist bei Mikrorechnern, die fuer Echtzeit - Aufgaben im Kfz eingesetzt werden, nicht moeglich, da die Rechenzeitbelastung aufgrund der hohen Rate zu uebertragender Botschaften unertraeglich waere.

3. Vorteile der Erfindung

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 1 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug moeglich ist.

Dies wird dadurch erreicht, dass die zu uebertragenden Botschaften sich bezueglich ihres Inhalts und ihrer Bedeutung mit Hilfe eines Identifizierers selbst identifizieren.

Besonders vorteilhaft ist es dabei, dem Identifizierer gleichzeitig eine Prioritaet zuzuordnen, wobei der Identifizierer dann vorzugsweise am Anfang der zu uebertragenden Botschaft steht.

Eine weitere vorteilhafte Weiterbildung besteht darin, dass der Identifizierer Inhalte wie Adressen, Daten, Sensorsignale,

Stellgroessen, Zwischenergebnisse, Befehle fuer Synchronisationen, Befehle zum Ausloesen von Aktionen, Drehzahl, Drehzahlgradienten, Motortemperatur, Last der Antriebsmaschine, Befehlsdaten wie z.B. von einer Anti-Blockier-Einrichtung oder einer Antischlupfeinrichtung, usw. kennzeichnet.

Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 1 angegebenen Verfahrens ergeben sich aus den Unteranspruechen, sowie aus der Zeichnung und der zugehoerigen Beschreibung.

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 7 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug moeglich ist.

Dies wird dadurch erreicht, dass die zu uebertragende Botschaft mit einem definierten Bitmuster anfaengt, dieses Bitmuster in eine Fehlererkennung einbezogen wird und auf das Kontrollwort der Fehlererkennung ein Bitmuster folgt, das zum Anfangsbitmuster komplementaer ist.

Besonders vorteilhaft ist es dabei, das Kontrollwort der Fehlererkennung durch die Multiplikation eines Generator - Polynoms mit dem Restklassenpolynom (X + 1) zu erzeugen.

Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 7 angegebenen Verfahrens ergeben sich aus den Unteranspruechen, sowie aus der Zeichnung und der zugehoerigen Beschreibung.

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 9 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug moeglich ist.

Dies wird dadurch erreicht, dass auf der die wenigstens zwei Rechner verbindenden Leitung dominante und rezessive Zustaende uebertragen werden und dass eine Fehlermeldung aus einer einzigen, mit der bei der Uebertragung der Botschaft auftretenden Zustandsfolge nicht zu verwechselnden Zustandsfolge dominanter Zustaende besteht.

Besonders vorteilhaft ist es dabei, dass die Fehlermeldung zu jedem beliebigen Zeitpunkt und von jedem Teilnehmer, der mit der Leitung gekoppelt ist, abgegeben werden kann. Weiter besteht ein Vorteil darin, dass das Ende der Fehlermeldung, bzw. das Ende der letzten Fehlermeldung zur zeitlichen Synchronisation aller Teilnehmer verwendet wird.

Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 9 angegebenen Verfahrens ergeben sich aus den Unteranspruechen, sowie aus der Zeichnung und der zugehoerigen Beschreibung.

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 16 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug moeglich ist.

Dies wird dadurch erreicht, dass jeder Teilnehmer alle Botschaften empfaengt, jedoch nur die fuer ihn relevante Botschaft weiterverarbeitet, wobei die relevanten Botschaften bei jedem Teilnehmer mit Hilfe einer Liste mit den empfangenen Botschaften in Beziehung gebracht werden.

Besonders vorteilhaft ist es dabei, dass die Reihenfolge der Liste fuer die interne Weiterverarbeitung im Teilnehmer prioritaet-bestimmend ist. Ein weiterer Vorteil besteht darin, dass die in den Botschaften enthaltenen Informationen in Abhaengigkeit in der jedem Teilnehmer vorliegenden Liste in Speicherzellen uebertragbar sind und dem Teilnehmer zur Weiterverarbeitung zur Verfuegung stehen.

Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 16 angegebenen Verfahrens ergeben sich aus den Unteranspruechen, sowie aus der Zeichnung und der zugehoerigen Beschreibung.

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 25 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung moeglich ist.

Dies wird dadurch erreicht, dass jeder Teilnehmer den zu uebertragenden Botschaften mittels einer Liste eine Prioritaet zuordnet und in jedem Zeitpunkt die Liste die Prioritaet der im Teilnehmer zur Uebertragung bereitstehenden Botschaften

bestimmt.

Besonders vorteilhaft ist es dabei, dass der Liste weitere Informationen insbesondere bezueglich des Uebertragungszustandes und bezueglich Uebertragungsanforderungen zugeordnet sind. Weiter ist es vorteilhaft, fehlerbehaftete Uebertragungen zu erfassen und zu registrieren und die Uebertragung bei aufgetretenen Fehlern zu wiederholen.

Bei einer weiteren vorteilhaften Weiterbildung handelt es sich darum, dass jedem Teilnehmer ein bestimmter maximaler Teil der Uebertragungskapazitaet der Leitung zugeordnet wird. Eine weitere vorteilhafte Moeglichkeit besteht darin, dass nach dem Auftreten einer bestimmten Anzahl von Fehlermeldungen eines bestimmten Teilnehmers sich dieser Teilnehmer sendesignalmaessig von der Leitung abkoppelt.
Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 25 angegebenen Verfahrens ergeben sich aus den Unteranspruechen sowie aus der Zeichnung und der zugehoerigen Beschreibung.

Das erfindungsgemaesse Verfahren mit den Merkmalen des Hauptanspruchs 33 hat gegenueber dem beschriebenen Stand der Technik den Vorteil, dass eine schnelle und sichere Datenuebertragung zwischen den im Kraftfahrzeug installierten Rechnern mit den besonderen Anforderungen einer Steuergeraete - Kopplung im Kraftfahrzeug moeglich ist.

Dies wird dadurch erreicht, dass bei dem erfindungsgemaessen Verfahren zur Uebertragung von zweiwertigen Informationen der eine Informationswert durch einen konstanten Signalpegel, der andere Informationswert mit alternierenden Signalpegeln uebertragen wird.

Besonders vorteilhaft ist es dabei, die alternierenden Signalpegel gleichzeitig zur Fehlererkennung heranzuziehen.

Weitere vorteilhafte Weiterbildungen und Verbesserungen des im Hauptanspruch 33 angegebenen Verfahrens ergeben sich aus den Unteranspruechen, sowie aus der Zeichnung und der zugehoerigen Beschreibung.

4. Ausfuehrungsbeispiel

4.1. Auflistung der einzelnen Figuren der Zeichnung

Fig. 1: Ausfuehrungsbeispiel fuer lineare Busstruktur

Fig. 2: Beispiel fuer galvanisch gekoppelte, asymmetrische Ansteuerung der Busleitung

Fig. 3: Beispiel fuer galvanisch gekoppelte, symmetrische Ansteuerung der Busleitung

Fig. 4: Beispiel fuer galvanisch getrennte, symmetrische Ansteuerung mit Uebertrager

Fig. 5: Ausfuehrungsbeispiel fuer Lichtleiter - Stern

Fig. 6: Ausfuehrungsbeispiel fuer Lichtleiter - Bus

Fig. 7: Aufbau einer Botschaft

Fig. 8: Aufbau CONTROL-FIELD

Fig. 9: Aufbau CRC-FIELD

Fig. 10: Aufbau einer Fehlermeldung

Fig. 11: Blockschaltbild des Schnittstellen-Bausteins

4.2. Physikalische Realisierung

Tabelle 1 zeigt die moeglichen Bustopologien, Ankopplungs- und Leitungsarten in Abhaengigkeit von dem gewaehlten Uebertragungsmedium.

Tabelle 1

Uebertragungs-medium	Bustopologie	Ankopplung	Leitungsart
Elektr. Leiter	linearer Bus Stern	galvanisch induktiv Optokoppler	Leitungspaar verdrilltes Leitungspaar Koaxialkabel
Lichtleiter	linearer Bus Stern mit zentr. Koppler	elektro-optische Wandler	Glasfasern Kunsstoff-fasern

Uebertragungsart: Zeitmultiplex
Basisbanduebertragung

Zugriffsart: Multimaster-Prinzip
deterministische Buszuteilung

raeumliche Ausdehnung: von Uebertragungsrate abhaengig

Das vorgestellte Bussystem kann genau dann realisiert werden, wenn auf dem Uebertragungsmedium zwei Zustaende mit den Eigenschaften dominant bzw. rezessiv moeglich sind.

3506118

Beispiele hierfuer sind:

	Parallel-Impedanz	Energie	Und-Gatter
dominant	niederohmig	vorhanden	0
rezessiv	hochohmig	nicht vorh.	1

Beispiele fuer Impedanz:

Schalter geschlossen / offen
Transistor leitend / nichtleitend

Beispiele fuer Energie :

Spannung liegt an / liegt nicht an
Licht an / aus

Fig. 1 zeigt die Ausfuehrung als lineares Bussystem. Das System besteht aus einer durchgehenden Busleitung (Adernpaar oder Lichtleiter) und Anschlussleitungen, die zu den einzelnen Teilnehmern gehen.

Fig. 2 zeigt eine Ausfuehrung mit galvanischer Ankopplung an die Busleitung. Die Treiber sind steuerbare Schalter, realisiert durch Transistoren (Open-Collector-Ankopplung). Als Busleitung dient ein Zweidrahtleitung oder ein Koaxkabel. Die Ansteuerung erfolgt asymmetrisch. Am Ende der Busleitung wird ueber Widerstaende eine Versorgungsspannung angelegt. Der dominante Buszustand liegt dann vor, wenn mindestens ein Treibertransistor leitet.

Fig. 3 zeigt eine Ausfuehrung, bei der als Sendetreiber zwei komplementaere Treibertransistoren eingesetzt werden, die gleichzeitig leitend bzw. sperrend gesteuert werden. Die Leitung wird dadurch symmetrisch angesteuert. Der Vorteil dieser Anordnung liegt in einer hoeheren Stoerfestigkeit und in einer niedrigeren Stoerabstrahlung.

Bei den oben gezeigten Ausfuehrungen ist die Busleitung galvanisch mit den einzelnen Stationen verbunden. Dadurch koennen in elektrisch bzw. in elektromagnetisch gestoerter Umgebung Probleme auftreten, wenn die Stationen ueber weitere Leitungen (z.B. Stromversorgungen) verkoppelt sind. Die folgenden Beispiele weisen diesen Nachteil nicht auf.

Fig. 4 zeigt eine Ausfuehrung, bei der zur galvanischen Trennung Uebertrager eingesetzt werden. Die Gleichstromfreiheit wird durch eine Biphase-Kodierung gewonnen. Der rezessive Buszustand wird durch Abkopplung aller Uebertrager von ihren Treibern erreicht (Treiber hochohmig schalten). Der dominante Buszustand wird durch das Aufschalten eines positiven oder eines negativen Impulses auf mindestens einen der Uebertrager erreicht. Die Synchronisierung der Impulspolaritaet geschieht ueber die Festlegung der Startbitpolaritaet.

Eine weitere Ausfuehrung mit galvanisch getrennter Ankopplung kann durch den Einsatz von Optokopplern erreicht werden. Besonders einfach wird dies durch den Einsatz von Optokopplern mit Open-Collector-Ausgaengen.

Fig. 5 zeigt als Beispiel fuer eine Ausfuehrung mit Lichtleitern ein Sternsystem mit einem zentralen optischen Koppler. Die Verkopplung kann passiv oder mit elektrooptischen Wandlern ausgefuehrt werden.
Dominanter Buszustand: mindestens eine Sendediode leuchtet.
Rezessiver Buszustand: alle Sendedioden dunkel.

Fig. 6. zeigt als Beispiel fuer eine Ausfuehrung mit Lichtleitern ein lineares Bussystem. Die Aschlussleitungen sind mit der zentralen Busleitung so verbunden, dass zur Ein- und Auskopplung von Licht keine aufwendigen passiven oder elektrooptischen Wandler benoetigt werden. Ein Beispiel ist die Verschmelzung von zentraler Busleitung und Anschlussleitung.
Dominanter Buszustand: mindestens eine Sendediode leuchtet.
Rezessiver Buszustand: alle Sendedioden dunkel.

4.3. Uebertragungsprotokoll

Eine Botschaft besteht aus den Bitfeldern START-OF-FRAME, IDENTIFIER, CONTROL-FIELD, DATA-FIELD, CRC-FIELD, ACK-FIELD, END-OF-FRAME und INTERMISSION (siehe Fig. 6)

HINWEIS:
In dieser Beschreibung werden die Begriffe HIGH und LOW im Sinne logischer Pegel verwendet.

Waehrend HIGH in seiner Wirkung auf dem Bus rezessiv ist, ist LOW dominant. Das hat zur Folge, dass von allen Busteilnehmern LOW empfangen wird, sofern mindestens einer von mehreren Busteilnehmern LOW sendet.

START-OF-FRAME

markiert den Botschaftsanfang und besteht aus einem einzigen LOW Bit.

Ein Busteilnehmer darf nur im Zustand BUS-IDLE (siehe Abschnitt 4.5), d.h. wenn der Bus frei ist, die Uebertragung einer Botschaft beginnen. Alle Empfaenger synchronisieren sich auf die fuehrende, durch START-OF-FRAME verursachte Flanke .

IDENTIFIER

kennzeichnet den Inhalt des Datenfeldes (DATA-FIELD) im Sinne eines Namens und einer Prioritaet.

Der IDENTIFIER hat nicht zwangslaeufig die Bedeutung einer Adresse, die einen einzigen Empfaenger aus einer Vielzahl von Busteilnehmern bestimmt. Ob eine korrekt empfangene Botschaft von den Busteilnehmern weiter bearbeitet wird oder nicht, wird ausschliesslich durch das AKZEPTANZ-FILTER (siehe Abschnitt 4.7.1.1.3.) eines jeden Busteilnehmers bestimmt. Aus der Sicht eines Senders gibt es deshalb keinen Unterschied zischen einer Punkt zu Punkt - Uebertragung und der gleichzeitigen Ansprache einiger oder aller Busteilnehmer.

Bei gleichzeitigem Buszugriff mehrerer Sender wird waehrend der Arbitrierungsphase anhand der Prioritaet bestimmt, welcher der Sender das Recht zur weiteren Uebertragung der Botschaft erhaelt (siehe Abschnitt 4.4.) .

Im vorliegenden Ausfuehrungsbeispiel ist der IDENTIFIER acht Bit lang.
(IFL = IDENTIFIER-FIELD-LENGTH = 8)

CONTROL-FIELD

umfasst die Bitfelder REMOTE-TRANSMISSION-REQUEST, DATA-BYTE-CODE und fuer zukuenftige Erweiterungen reservierte Bits.

REMOTE-TRANSMISSION-REQUEST zeigt an, ob durch die entsprechende Botschaft Daten uebertragen oder angefordert werden sollen.DATA-BYTE-CODE enthaelt Information ueber die Laenge des Datenfeldes.

In dem vorliegenden Ausfuehrungsbeispiel ist das CONTROL-FIELD acht Bit lang.
(CFL = CONTROL-FIELD-LENGTH = 8)

DATA-FIELD

enthaelt die zu uebertragende Information, deren Bedeutung durch den IDENTIFIER festgelegt ist. Die Laenge dieses Feldes (DATA-BYTE-COUNT) ist variabel und kann zB. ein, zwei, vier oder acht Bytes betragen.

DATA-BYTE-COUNT = 2 ** DATA-BYTE-CODE
DFL = DATA-FIELD-LENGTH = 8 * DATA-BYTE-COUNT

CRC-FIELD

Dieses Bitfeld enthaelt das mittels eines Generator - Polynoms erzeugte Kontrollwort (CRC-SEQUENCE), es wird mit einem CRC-DELIMITER abgeschlossen (Fig. 9).

Das Generator - Polynom ist fuer kleine Botschaftslaengen (kleiner 120 zu sichernde Bits) ausgewaehlt, womit eine hoehere Hamming - Distanz erreicht wird wie bei der Absicherung langer Botschaften mit der gleichen Anzahl von Kontrollbits.

In dem vorliegenden Ausfuehrungsbeispiel ist die CRC-SEQUENCE 15 Bits lang.
(CRCL = CRC-SEQUENCE-LENGTH = 15)

Durch das Kontrollwort werden die Felder START-OF-FRAME, IDENTIFIER, CONTROL-FIELD, DATA-FIELD und CRC-SEQUENCE (ohne CRC-DELIMITER) gesichert.

CRC-DELIMITER

Der CRC-DELIMITER besteht aus einem HIGH Bit, er folgt auf das CRC - Kontrollwort und schliesst das CRC-FIELD ab.

Zyklische Codes koennen solche Fehler nicht aufdecken, die sich auf die zyklische Verschiebung des gesamten Codewortes (zu sichernde Bits und Sicherungsbits) zurueckfuehren lassen. Eine zyklische Verschiebung ergibt naemlich wieder ein gueltiges Codewort.

Wird jedoch das Bit START-OF-FRAME, das per Definition LOW ist, in das Codewort einbezogen, kann jede Rotation des Codewortes daran erkannt werden, dass das CRC-FIELD nicht mit HIGH abgeschlossen wird.

ACK-FIELD

Alle Empfaenger teilen dem Sender den Empfang einer korrekten Botschaft dadurch mit, dass sie als erstes Bit in diesem Fenster ein LOW - Bit uebertragen. Als zweites Bit wird HIGH (ACK-DELIMITER) uebertragen. Der Sender kann auf diese Art und Weise erkennen, ob zumindestens einer der Busteilnehmer die Botschaft fehlerfrei empfangen hat.

Alle diejenigen Busteilnehmer, die eine fehlerhafte Botschaft empfangen haben, melden dies mittels eines ERROR-FLAG.

Erhaelt der Sender keine Bestaetigung fuer den korrekten Empfang seiner Botschaft durch zumindest einen Empfaenger, so setzt er selbst eine Fehlermeldung ab.

END-OF-FRAME

besteht aus einer ununterbrochenen Folge von HIGH Bits und steht am Ende einer jeden Botschaft.

Die Laenge von END-OF-FRAME ist so zu waehlen, dass all diejenigen Empfaenger, die aufgrund einer fehlerhaften Uebertragung der Laengeninformation an dieser Stelle Daten

erwarten, bereits beim zweitletzten Bit von END-OF-FRAME einen Stuffehler feststellen (siehe KODIERUNG). Unabhaengig von vorangegangenen Uebertragungsfehlern kann so jeder Busteilnehmer das Ende einer Botschaft erkennen.

Tastet der Sender waehrend END-OF-FRAME zumindest ein LOW Bit (zB. Bit von ERROR-FLAG) auf dem Bus ab, so wird dies von dem zuletzt aktiven Sender als Reaktion auf die soeben von ihm uebertragene Botschaft angesehen. Dadurch besteht eine eindeutige Zuordnung von Fehlermeldungen, auch wenn der Fehlerzustand erst mit der Uebertragung des zweitletzten Bits einer Botschaft durch einen der Empfaenger erkannt wird.

In dem vorliegenden Ausfuehrungsbeispiel ist END-OF-FRAME sieben Bits lang.
(EOFL = END-OF-FRAME-LENGTH = 7)

INTERMISSION

besteht aus einer ununterbrochenen Folge von HIGH Bits. In dieser Zeit darf keiner der Busteilnehmer die Uebertragung einer neuen Botschaft beginnen.

Waehrend INTERMISSION hat jeder Empfaenger die Moeglichkeit, eine Ueberlastung (fehlende Empfangsbereitschaft) durch Senden eines ERROR-FLAG zu melden, so dass das Absenden der naechsten Botschaft verzoegert wird, bis alle Empfaenger wieder bereit sind.

Eine in INTERMISSION fallende Fehlermeldung wird genauso wie die in andere Bitfelder fallenden Fehlermeldungen behandelt. Eine Wiederholung der vorangegangenen Botschaft durch den entsprechenden Sender erfolgt jedoch nicht.

In dem vorliegenden Ausfuehrungsbeispiel ist INTERMISSION drei Bit lang.
(IML = INTERMISSION-LENGTH = 3)

BUS-IDLE

Der Bus ist frei, wenn er nach Ablauf von INTERMISSION weiterhin HIGH - Pegel fuehrt. Der Zustand BUS-IDLE kann von beliebiger Dauer sein. Der Empfang eines LOW Bits wird als START-OF-FRAME interpretiert.

Eine Fehlermeldung besteht aus dem Bitfeldern ERROR-FLAG und ERROR-DELIMITER (siehe Fig. 10)

ERROR-FLAG

besteht aus einer ununterbrochenen Folge von LOW-Bits.

Die Laenge der Folge ist so zu bestimmen, dass durch sie das Gesetz des 'Bitstuffing' verletzt wird, das auf alle Bitfelder von START-OF-FRAME bis CRC-DELIMITER angewendet wird. Die anderen Busteilnehmer reagieren darauf, indem sie selbst eine Fehlermeldung senden.

In dem vorliegenden Ausfuehrungsbeispiel ist das ERROR-FLAG sechs Bit lang.
(EFL = ERROR-FLAG-LENGTH = 6)

ERROR-DELIMITER

Jede Fehlermeldung wird mit einer ununterbrochenen Folge von HIGH Bits abgeschlossen.

Der Sendevorgang des ERROR-DELIMITER beginnt, nachdem auch alle anderen Teilnehmer mit der Uebertragung des ERROR-FLAG fertig sind (LOW nach HIGH Uebergang auf dem Bus).

Im vorliegenden Ausfuehrungsbeispiel ist der ERROR-DELIMITER sieben Bit lang.
(EDL = ERROR-DELIMITER-LENGTH = 7)

4.4. Bus-Organisation

Die Organisation des Busverkehrs beruht auf folgenden fuenf Regeln:

(1) BUS-ZUGRIFF

Die Busteilnehmer duerfen nur im Zustand BUS-IDLE die Uebertragung einer Botschaft starten.

Alle Empfaenger muessen auf die fuehrende Flanke von START-OF-FRAME synchronisieren.

(2) ARBITRIERUNG

Beginnen zwei oder mehr Busteilnehmer gleichzeitig mit der Uebertragung, wird der Zugriffskonflikt durch einen Arbitrierungsmechanismus auf Bitebene geloest.

Waehrend der Arbitrierung vergleicht jeder Sender den Bitpegel, den er selbst auf den Bus schreibt, mit demjenigen, den er tatsaechlich auf dem Bus abtastet. Alle diejenigen Sender, die nicht den von ihnen selbst gesendeten Buspegel vorfinden, muessen die Uebertragung ohne auch nur ein weiteres Bit zu senden einstellen. Durch diese Vereinbarung kann der Arbitrierungsvorgang ablaufen, ohne dass irgendwelche Information auf dem Bus zerstoert wird. Derjenige Sender, dessen Botschaft den IDENTIFIER mit der hoechsten Prioritaet aufweist, gewinnt den Buszugriff. Er uebertraegt die begonnene Botschaft zu Ende.

(3) KODIERUNG

Die Botschaftssegmente START-OF-FIELD, IDENTIFIER, CONTROL-FIELD, DATA-FIELD und CRC-FIELD werden nach der Methode des Bitstuffing kodiert. Treten in dem zu uebertragenden Datenstrom in ununterbrochener Reihenfolge fuenf gleiche Bits auf, so fuegt der Sender automatisch ein Bit mit umgekehrter Wertigkeit in den Bitstrom ein.

(4) DEKODIERUNG

Die Botschaftssegmente START-OF-FIELD, IDENTIFIER, CONTROL-FIELD, DATA-FIELD und CRC-FIELD werden nach der Methode des Bitstuffing kodiert. Empfaengt ein Busteilnehmer fuenf gleiche Bitpegel in ununterbrochener Reihenfolge, so entfernt dieser das nachfolgende Stuffbit aus dem Bitstrom. Die Wertigkeit des Stopfbits muss invers zu der der voran gegangenen Bits sein (Fehlerpruefung).

(5) FEHLERMELDUNG

Jeder Busteilnehmer, der einen Uebertragungsfehler feststellt, teilt dies allen anderen Busteilnehmern mit, indem er sechs aufeinander folgende LOW Bits (ERROR-FLAG) sendet.

Die Fehlermeldung erfolgt sofort nach einem erkannten Fehler, dh. beginnend mit dem auf den Fehlerzustand folgenden Bit. Nur bei Erkennung eines CRC-Fehlers wird die Fehlermeldung erst drei Takte spaeter abgeschickt. Dadurch wird sichergestellt, dass das ACK-FIELD nicht durch eine von einem CRC - Fehler ausgeloeste Fehlermeldung ueberschrieben wird.

Die Fehlermeldung wird durch einen Uebergang von LOW nach HIGH nach mindestens sechs LOW Bits auf dem Bus beendet (siehe Zustandsdiagramm zur Fehlerbehandlung)

(6) UEBERLAST

Bei Ueberlastung (empfangene Botschaft ist noch nicht bearbeitet) hat jeder Busteilnehmer die Moeglichtkeit, durch Senden eines ERROR-FLAG waehrend INTERMISSION dies allen anderen Teilnehmern zu melden (siehe INTERMISSION).

4.5. Zustandsdiagramme

4.5.1. Erlaeuterungen zu den Zustandsgraphen

Jeder Rechteckrahmen stellt einen Systemzustand dar. Die Uebergaenge zwischen den Zustaenden, die von Eingangs- und Zustandsvariablen abhaengen, sind durch gerichtete Verbindungslinien dargestellt. Die Zustandsuebergaenge erfolgen stets mit der aktiven Flanke des Bustaktes. Mit der gleichen Taktflanke werden Ausgangsdaten auf die Busleitung gelegt. Der tatsaechliche Buspegel wird erst kurz vor der naechsten aktiven Bustaktflanke abgetastet.

Nach einem BUS-IDLE Zustand wird der Bustakt auf den naechsten Buspegeluebergang von HIGH nach LOW synchronisiert.

Die Eingangs- und Zustandsvariablen koennen geordnet und zu einem Entscheidungsvektor zusammengefasst werden.

Der Entscheidungsvektor hat folgende Form:

(BUS-MONITOR, BUS-DRIVE, STUFF, COUNT,
TX-REQUEST, CRC-ERROR, OVERLOAD)

Es bedeutet:

BUS-MONITOR

Eingangsvariable, die den auf der Busleitung abgetasteten Logikpegel wiederspiegelt.

BUS-MONITOR = 0 entspr. dominantem Buspegel (LOW)
BUS-MONITOR = 1 entspr. rezessivem Buspegel (HIGH)

BUS-DRIVE

Zustandsvariable, die den Logikpegel angibt, der in diesem Zustand gesendet wird.

BUS-DRIVE = 0 Sendepegel dominant (LOW)
BUS-DRIVE = 1 Sendepegel rezessiv (HIGH)

STUFF

Eingangsvariable, die angibt, ob der Buspegel waehrend der letzten fuenf Abtasttakte konstant war. (fuenfmal LOW oder fuenfmal HIGH). Der Pegel, den BUS-MONITOR angibt, ist der aktuellste Wert dieser Fuenferkette.

STUFF = 0 Buspegel hat gewechselt
STUFF = 1 Buspegel war konstant

COUNT

Zustandsvariable, die angibt, ob der interne Zaehler (CNTR) abgelaufen ist. Dieser Zaehler wird nicht in allen Zustaenden benoetigt. Falls benoetigt, wird der Zaehler beim Uebergang in den entsprechenden Zustand gesetzt.

COUNT = 0 Zaehler noch nicht abgelaufen (CNTR <> 0)
COUNT = 1 Zaehler abgelaufen (CNTR = 0)

TX-REQUEST

Eingangsvariable, die angibt, ob eine Botschaft zum Absenden bereit liegt, so dass an der naechsten Buszuteilungsprozedur teilgenommen werden muss.

TX-REQUEST = 0 Sendeauftrag liegt nicht vor
TX-REQUEST = 1 Sendeauftrag liegt vor

CRC-ERROR

Zustandsvariable, die angibt, ob in der empfangenen Botschaft ein Uebertragungsfehler vorlag. Die Variable wird nach dem Empfang des letzten Bits der CRC-SEQUENCE gesetzt.

CRC-ERROR = 0 Empfang war fehlerfrei
CRC-ERROR = 1 Uebertragungsfehler

OVERLOAD

Zustandsvariable die angibt, ob die letzte empfangene Botschaft von der Schnittstellenlogik aufgearbeitet werden konnte oder ob diese ueberlastet ist.

OVERLOAD = 0 Empfangslogik ueberlastet
OVERLOAD = 1 Empfangslogik bereit

Mit dem Entscheidungsvektor ist eindeutig festgelegt, in welchen Folgezustand gegangen wird. Der Zustandsgraph kann deshalb mit Hilfe der Entscheidungsvektoren beschrieben werden. Ist fuer ein bestimmtes Element im Zustandsvektor ein 'x' angegeben, so bedeutet dies, dass die entsprechende Variable

fuer die Entscheidung nicht relevant ist (die Variable darf '0' oder '1' sein).

4.5.2. Beschreibungsbeispiel fuer den Zustand BUS-IDLE

Der Zustand BUS-IDLE ist der regulaere Folgezustand des Zustands TEST-INTERMISSION. Im Zustand BUS-IDLE wird stets der rezessive Sendepegel HIGH auf die Busleitung gelegt.

Uebergaenge in Folgezustaende:

1) Entscheidungsvektor (1;1;1;x;0;x;x):

   Dieser Entscheidungsvektor bedeutet folgendes:
   Buspegel HIGH; Sendepegel HIGH; Bus war schon mind. fuenf Takte lang HIGH; Zaehler nicht relevant; keine Sendeanforderung; Uebertragungsfehler und Ueberlastung nicht relevant. Da keine Busaktion detektiert wurde und keine Sendeanforderung vorlag; ist der Folgezustand wieder BUS-IDLE.

2) Entscheidungsvektor (1;1;1;x;1;x;x):

   Jetzt liegt bei sonst gleichen Variablen wie in 1) eine Sendeanforderung vor. Da die Busleitung frei ist; kann sofort mit dem Senden begonnen werden. Der Folgezustand ist also TRANSMIT-START-OF-FRAME.

3) Entscheidungsvektor (0;1;0;x;x;x;x):

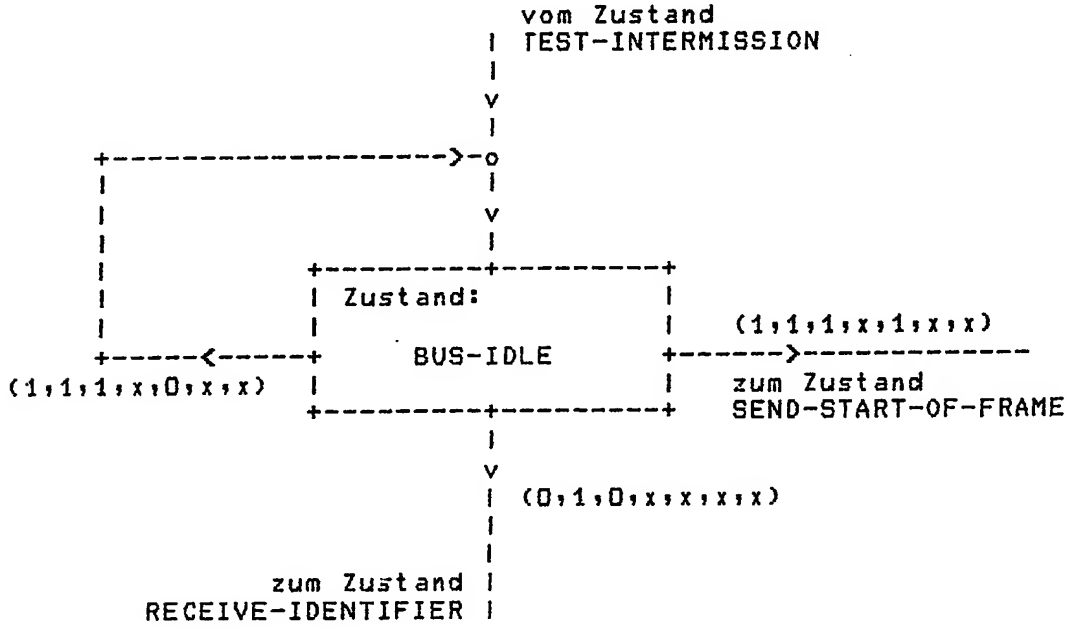
   Auf der Busleitung wurde der Pegel LOW detektiert; das heisst; dass mindestens ein anderer Busteilnehmer begonnen hat; eine Botschaft zu uebertragen. Das detektierte LOW-Bit ist START-OF-FRAME. Der Folgezustand ist RECEIVE-IDENTIFIER.

4) Alle weiteren Entscheidungsvektoren deuten auf eine Hardware-Stoerung bzw. auf einen Hardware-Ausfall innerhalb des Schnittstellenbausteins hin.

Entsprechendes gilt fuer alle anderen Systemzustaende; die in den Zustandsgraphen EMPFANGSMODUS; SENDEMODUS und FEHLER-BEHANDLUNG beschrieben werden.

Neben den Entscheidungsvektoren; die sich aufgrund von

Stoerungen auf der Busleitung ergeben, werden nur die Vektoren aufgefuehrt, die bei funktionsfaehiger Hardware auftreten koennen. Die restlichen Vektoren kann man entweder zur Vereinfachung des Steuerwerks ausnuetzten oder zur Erhoehung der Systemsicherheit in einem Zustand HARDWARE-ERROR behandeln.



4.5.3. Zustandsdiagramm EMPFANGS-MODUS

Aus dem Zustand BUS-IDLE kommt man durch Erkennen von START-OF-FRAME auf dem Bus in den Zustand RECEIVE-IDENTIFIER. Beim Zustandsuebergang wird der Zaehler CNTR mit IFL=8 (IDENTIFIER-FIELD-LENGTH) vorbelegt. IFL gibt an wieviel Kennungsbits vereinbart wurden (siehe 4.3, IDENTIFIER).

Mit jedem empfangegen Kennungsbit wird die Schleife des Zustands RECEIVE-IDENTIFIER einmal durchlaufen und der Zaehler wird dekrementiert. Tritt im IDENTIFIER ein Stuffbit auf, so erfolgt aufgrund von STUFF = 1 der Uebergang in den Zustand

IGNORE-ID-STUFFB . Dort wird das Stuffbit ausgeblendet. Ist danach immernoch STUFF = 1, so muss eine Ausnahmesituation vorliegen (Stoerung auf Busleitung, Fehlermeldung eines anderen Teilnehmers, unerwartete Busruhe); als Reaktion darauf wird in den Zustand ERROR-HANDLING gegangen. Ist jedoch STUFF = 0, so kann der Ruecksprung in den Zustand RECEIVE-IDENTIFIER erfolgen.

Nach Ablaufen des Zaehlers geht das System (entweder aus dem Zustand RECEIVE-IDENTIFIER oder IGNORE-ID-STUFFB) in den Zustand RECEIVE-CONTROL-FIELD ueber. Hier wird der Zaehler mit CFL=8 (CONTROL- FIELD-LENGTH) vorbelegt.

Sind alle Bits des CONTROL-FIELDs empfangen, so wird als naechstes in Zustand RECEIVE-DATA-FIELD das Datenfeld empfangen. Als Besonderheit gilt hier, dass der Zaehler nicht mit einer Konstanten, sondern mit der Variablen DFL (DATA-FIELD-LENGTH) vorbelegt wird. DFL kann die Werte 8, 16, 32 oder 64 annehmen.

Als naechstes wird im Zustand RECEIVE-CRC-SEQUENCE das CRC-Kontrollwort (CRC-SEQUENCE) empfangen. Hierzu wird zunaechst der Zaehler CNTR mit CRCL=15 (CRC-SEQUENCE-LENGTH) vorbelegt. Nach dem Empfang des letzten CRC-Bits wird die Zustandsvariable CRC-ERROR gesetzt (CRC-ERROR = 0 bedeutet kein Fehler, CRC-ERROR = 1 bedeutet Fehler im Uebertragungsrahmen).

Das CRC-Kontrollwort muss stets durch ein Bit (CRC-DELIMITER) mit dem Pegel HIGH abgeschlossen sein. Dies wird im Zustand RECEIVE-CRC-DELIMITER geprueft. Bei falschem Buspegel wird zum Zustand ERROR-HANDLING gesprungen.

Im naechsten Bustaktzyklus muessen die Empfaenger den Empfang der Botschaft bestaetigen. Dies geschieht im Zustand SEND-ACKNOWLEDGE durch Senden eines LOW-Pegels fuer fehlerfreien Empfang bzw. durch Senden eines HIGH-Pegels im Fehlerfall. Beim Entdecken des Buspegels HIGH wird, falls der dominante Pegel LOW gesendet wurde, zum Zustand ERROR-HANDLING uebergegangen.

Auf das erste Acknowledge-Bit folgt der ACK-DELIMITER, der stets den Pegel HIGH haben muss. Dieser Pegel wird im Zustand SEND-ACK-DELIMITER von allen Empfaengern ausgegeben. Wird der Pegel LOW empfangen, so wird zum Zustand ERROR-HANDLING uebergegangen.

Beim Zustandsuebergang nach RECEIVE-END-OF-FRAME wird der Zaehler CNTR mit EOFL=7 (END-OF-FRAME-LENGTH) initialisiert. Falls CRC-ERROR = 1 ist, oder falls waehrend

RECEIVE-END-OF-FRAME der Buspegel LOW empfangen wird, wird zum Zustand ERROR-HANDLING uebergegangen.

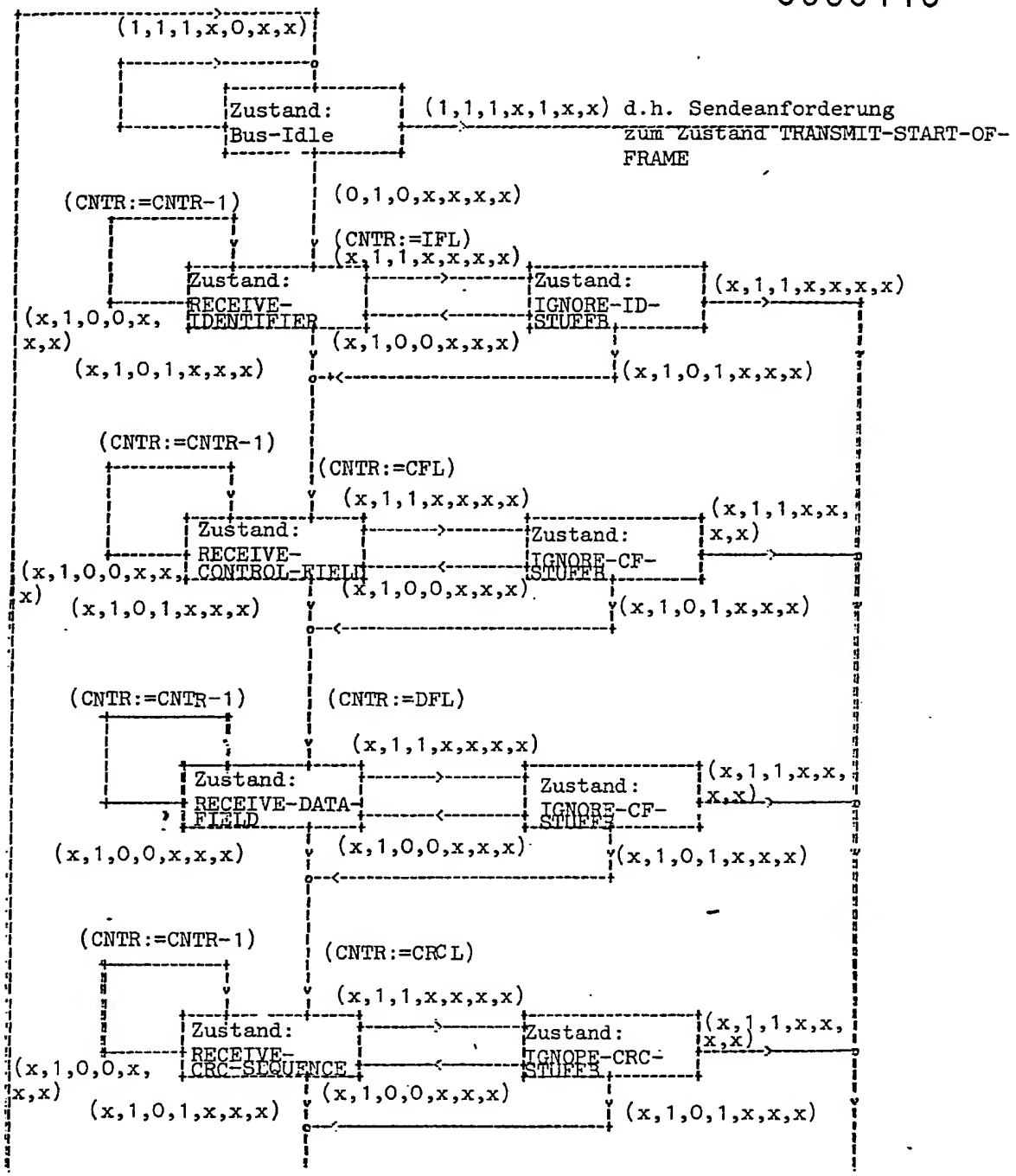
Im fehlerfreien Fall wurde die Botschaft jetzt vollstaendig empfangen. Als naechstes folgt der Zustand TEST-INTERMISSION, der aus IML=3 (INTERMISSION-LENGTH) Zyklen besteht. Der Buspegel muss dauernd HIGH sein, sonst wird nach ERROR-HANDLING gesprungen. Im Zustand TEST-INTERMISSION hat jeder Empfaenger die Moeglichkeit eine Ueberlastung (OVERLOAD = 1) zu melden, sodass das Absenden der naechsten Botschaft verzoegert wird, bis der Empfaenger wieder bereit ist. Auch dies geschieht durch ERROR-HANDLING. Nach Ablauf des Zaehlers CNTR wird in den Folgezustand BUS-IDLE uebergegangen, mit dem ein neuer Empfangs- oder Sendezyklus beginnt.

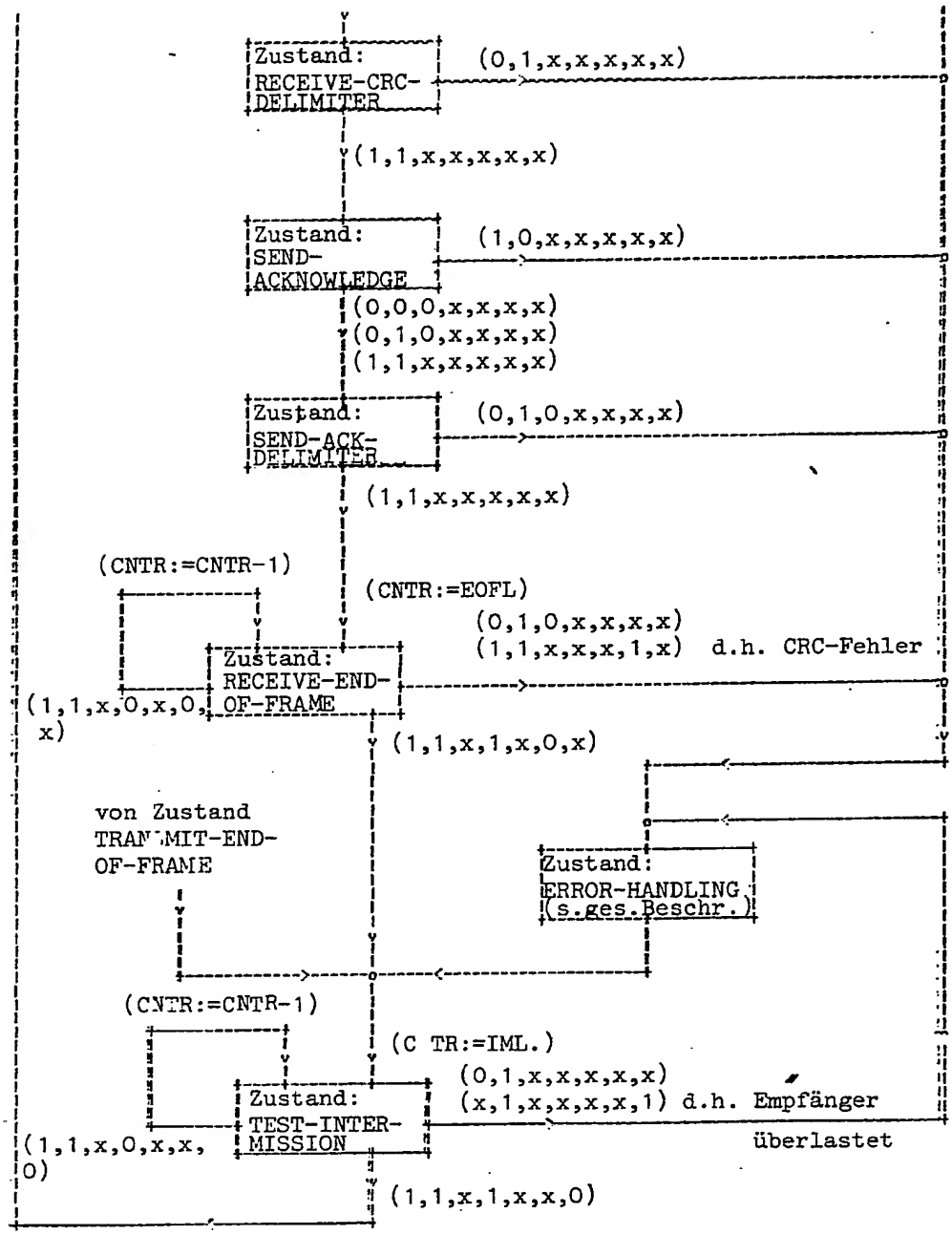
Bezueglich der Stuffbits, die in den Feldern CONTROL-FIELD, DATA-FIELD und CRC-SEQUENCE auftreten koennen, gelten die zum Zustand RECEIVE-IDENTIFIER gemachten Ausfuehrungen entsprechend.

Empfangs-Modus

Entscheidungsvektor: (Bus-Monitor, Bus-Drive, Stuff, Count, TX-Request, CRC-Error, Overload)

3506118





4.5.4. Zustandsdiagramm SENDE-MODUS

Vom Zustand BUS-IDLE aus erfolgt der Uebergang in den Sendemodus dann, wenn vor oder waehrend BUS-IDLE eine Sendeanforderung eintrifft (TX-REQUEST = 1). Siehe hierzu Zustandsgraph RECEIVE-MODE.

Das Senden einer Botschaft beginnt mit dem Zustand TRANSMIT-START-OF-FRAME, in dem der dominante Sendepegel LOW ausgegeben wird. Falls eine Stoerung den dominanten Pegel LOW in HIGH verfaelscht, wird zum Zustand ERROR-HANDLING uebergegangen.

Sonst folgt im Zustand TRANSMIT-IDENTIFIER die Buszuteilungsprozedur. Am Anfang wird der Zaehler auf IFL=8 (IDENTIFIER-FIELD-LENGTH) gesetzt.

Im Zustand TRANSMIT_IDENTIFIER wird verblieben (lediglich der Zaehler wird dekrementiert), wenn der empfangene Pegel dem gesendeten entspricht und die letzten fuenf Buspegel nicht konstant waren und der Zaehler noch nicht abgelaufen ist.

Der Zustand TRANSMIT-IDENTIFIER wird verlassen, wenn

- der gesendete rezessive HIGH-Pegel durch LOW ueberschrieben wurde. Die Buszuteilung ist damit verloren.
  Falls STUFF = 1 ist, ist der Folgezustand IGNORE-ID-STUFFB. Sonst ist bei nicht abgelaufenem Zaehler der Folgezustand RECEIVE-IDENTIFIER, bei abgelaufenem Zaehler RECEIVE-CONTROL-FIELD.

- ein gesendeter dominanter LOW-Pegel in HIGH verfaelscht wird. Der Folgezustand ist ERROR-HANDLING.

- der empfangene Pegel dem gesendeten entspricht. Falls die letzten fuenf Pegel gleich waren muss in den Zustand INSERT-ID-STUFFB uebergegangen werden. Sonst wird bei abgelaufenem Zaehler in den Folgezustand TRANSMIT-CONTROL-FIELD uebergegangen. Im letzten Fall wurde die Buszuteilung gewonnen.

Im Zustand INSERT-ID-STUFFB werden die Stuffbits des IDENTIFIER-FIELDS eingefuegt. Da in diesem Zustand die Buszuteilung nicht verloren werden kann, ist im Falle von unterschiedlichen Sende- und Empfangs- Pegeln ERROR-HANDLING der Folgezustand. Ansonsten wird bei nicht abgelaufenem Zaehler in den Zustand TRANSMIT-IDENTIFIER zurueckgegangen, bei abgelaufenen Zaehler wurde die Buszuteilung gewonnen, der Folgezustand ist TRANSMIT-CONTROL-FIELD.

Beim Einstieg in den Zustand TRANSMIT-CONTROL-FIELD wird der Zaehler CNTR mit CFL=8 initialisiert (CONTROL-FIELD-LENGTH). Da

die Buszuteilung beendet ist, muss ab jetzt der Sende- und Empfangs-Pegel stets gleich sein. Ist dies nicht der Fall, so wird in den Zustand ERROR-HANDLING uebergegangen. Bei nicht abgelaufenem Zaehler und bei Flankenwechsel innerhalb der letzten fuenf Taktzyklen bleibt das System im Zustand TRANSMIT-CONTROL-FIELD, lediglich der Zaehler wird dekrementiert. Das Einfuegen von Stuffbits geschieht im Zustand INSERT-CF-STUFFB. Bei abgelaufenem Zaehler erfolgt der Uebergang in den Folgezustand TRANSMIT-DATA-FIELD.

Die Vorgaenge im Zustand TRANSMIT-CONTROL-FIELD gelten auch fuer die Zustaende TRANSMIT-DATA-FIELD und TRANSMIT-CRC-SEQUENCE. Beim Einstieg in TRANSMIT-DATA-FIELD wird der Zaehler CNTR mit der Variablen DFL (DATA-FIELD-LENGTH) vorbelegt, die die Werte 8,16, 32 oder 64 annehmen kann. Beim Einstieg in TRANSMIT-CRC-SEQUENCE wird der Zaehler mit der Konstanten CRCL=15 (CRC-FIELD-LENGTH) vorbelegt.

Im Anschluss an die CRC-SEQUENCE muss der CRC-DELIMITER uebertragen werden. Dies geschieht im Zustand TRANSMIT-CRC-DELIMITER. Falls der gesendete HIGH-Zustand nach LOW verfaelscht wird, wird in den Zustand ERROR-HANDLING uebergegangen.

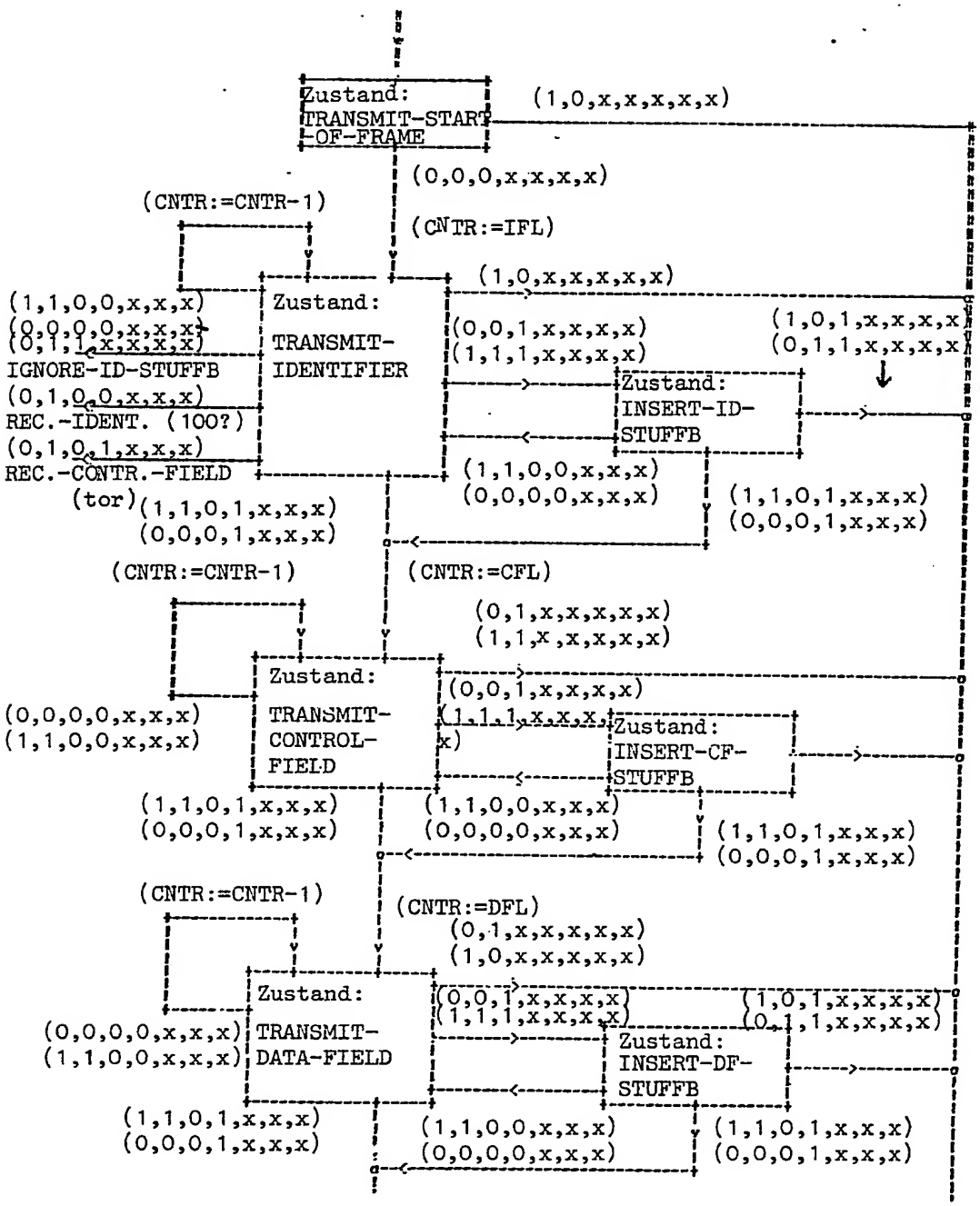
Als naechstes wird das Acknowledge-Bit geprueft. Ein empfangener HIGH-Pegel bedeutet, dass kein Teilnehmer die Botschaft fehlerfrei empfangen hat oder dass alle Teilnehmer eine falsche Rahmensynchronisation haben. Der Sender gibt deshalb im Zustand ERROR-HANDLING eine Fehlermeldung. Bei empfangenem LOW-Pegel wird in den Folgezustand RECEIVE-ACK-DELIMITER uebergegangen. Der ACK-DELIMITER muss stets HIGH sein. Ist dies nicht der Fall, so wird nach ERROR-HANDLING gesprungen.

Das Botschaftsende wird im Zustand TRANSMIT-END-OF-FRAME uebertragen. Da END-OF-FRAME aus sieben HIGH-Bits besteht, wird der Zaehler CNTR mit EOFL=7 (END-OF-FRAME-LENGTH) vorbelegt. Bei empfangenem HIGH-Pegel bleibt das System im Zustand TRANSMIT-END-OF-FRAME, bis der Zaehler abgelaufen ist und geht dann zu TEST-INTERMISSION ueber. Bei empfangenem LOW-Pegel wird dagegen nach ERROR-HANDLING gesprungen.

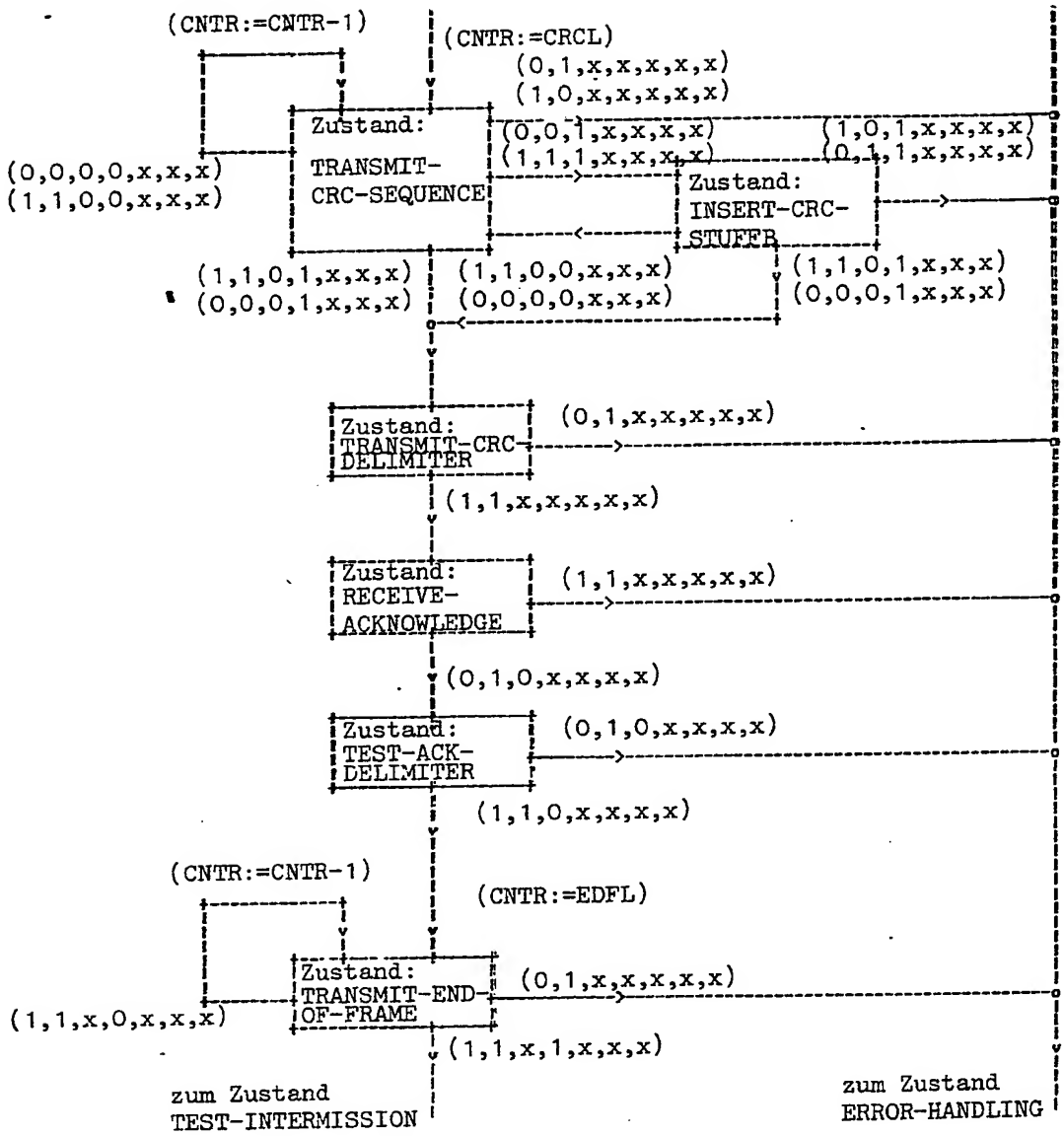
Mit dem Uebergang in den Zustand TEST-INTERMISSION ist der TRANSMIT-MODE beendet. Der Zustand TEST-INTERMISSION ist mit dem im RECEIVE-MODE beschriebenen identisch.

SENDE-MODUS

Entscheidungsvektor: (BUS-MONITOR, BUS-DRIVE, STUFF, COUNT, TX-Request, CRC-ERROR, OUERLOAD)



10.10.85
3506118



4.5.5. Zustandsdiagramm FEHLERBEHANDLUNG

Die Fehlerbehandlung beginnt mit dem Zustand SEND-ERROR-FLAG. Das ERROR-FLAG dient dazu, allen Teilnehmern mitzuteilen, dass auf dem Bus ein Uebertragungsfehler stattgefunden hat oder dass ein Empfaenger ueberlastet ist. Das ERROR-FLAG besteht aus sechs aufeinanderfolgenden LOW-Bits, deshalb wird der Zaehler CNTR beim Einstieg in SEND-ERROR-FLAG auf EFL=6 (ERROR-FLAG-LENGTH) gesetzt. Falls eine Stoerung die gesendeten dominanten Pegel LOW nach HIGH verfaelscht, wird noch einmal mit ERROR-HANDLING begonnen. Sonst wird der Zaehler dekrementiert. Beim Zaehlerstand Null erfolgt der Uebergang zum Zustand ERROR-SYNC.

Im Zustand ERROR-SYNC wird gewartet, bis andere Teilnehmer, die eventuell auch ein ERROR-FLAG senden, fertig sind. Dies wird daran erkannt, dass der Buspegel nach HIGH geht.

Die Fehlerbehandlung wird durch das Senden des ERROR-DELIMITERs im Zustand SEND-ERROR-DELIMITER abgeschlossen. Beim Einstieg wird der Zaehler CNTR mit EDL-1=6 (ERROR-DELIMITER-LENGTH = 7) vorbelegt. Der ERROR-DELIMITER besteht aus sieben HIGH-Bits, das erste dieser Bits wurde jedoch bereits im Zustand ERROR-SYNC gesendet. Wird beim Senden der restlichen Bits auf dem Bus ein LOW-Pegel entdeckt, so wird erneut mit ERROR-HANDLING begonnen. Sonst wird nach dem Ablaufen des Zaehlers zum Zustand TEST-INTERMISSION weitergegangen. Damit ist die Fehlerbehandlung beendet.

10.10.85
3506118

4.6. Fehlerbehandlung

4.6.1. Fehlereaktion

Jeder Teilnehmer sendet sofort nach einem erkannten Fehler, dh. beginnend mit dem auf den Fehlerzustand folgenden Bit, seine Fehlermeldung (ERROR-FLAG). Nur bei Erkennung eines CRC-Fehlers wird die Fehlermeldung erst drei Takte spaeter abgeschickt. Dadurch wird sichergestellt, dass das ACK-FIELD nicht durch eine von einem CRC - Fehler ausgeloeste Fehlermeldung ueberschrieben wird. Durch das Freihalten des ACK-FIELD kann ein Sender sowohl eine Bestaetigung seiner Botschaft von einigen Empfaengern erhalten als auch eine Fehlermeldung von den restlichen Empfaengern. Dies bietet die Moeglichkeit, im Wiederholungsfall mit hoher Wahrscheinlichkeit festzustellen, ob der Ausfall beim Teilnehmer (Sender) selbst oder bei einem Empfaenger liegt.

Eine Fehlermeldung besteht aus mindestens sechs aufeinanderfolgenden LOW Bits (ERROR-FLAG). Nach dem anschliessenden LOW - HIGH Uebergang muessen noch mindestens 10 HIGH Bits abgewartet werden (ERROR-DELIMITER un INTERMISSION).

4.6.2. Fehlerauftreten

Die Fehlerbehandlung kann sowohl von ihrem raeumlichen als auch ihrem zeitlichen Auftreten abhaengen.

Unter 4.6.5. werden einige Beispiele von Fehlerreaktionen des Open-Kollektor Busses gezeigt.

Aus der folgenden Tabelle 4.6.2 koennen alle moeglichen Einzelbit-Fehlerfaelle entnommen werden.

Unter den angegebenen Nummern sind die hier unter 4.6.5. aufgefuehrten Beispiele fuer Fehlerfaelle zu finden.

Tabelle 4.6.2

zeitliches Auftreten	raeumliches	am Sender und allen Empfaeng.	am Sender und einem Teil der Empfaeng	am Sender und kein Empfaenger	nicht am Sender aber allen Empfaengern	nicht am Sender aber Teil der Empfaeng.
BUS-IDLE				4.6.5.4.		4.6.5.7.
START-OF-FRAME						
IDENTIFIER		4.6.5.1.				
CONTROL-FIELD (DATA-BYTE-COUNT)					4.6.5.6.	
DATA-FIELD			4.6.5.2.	4.6.5.5.		4.6.5.8.
CRC-FIELD CRC-DELIMITER						
ACK-SLOT			4.6.5.3.			
END-OF-BLOCK						
INTERMISSION						

4.6.3. Fehlerklassen

Auf Grund der in 4.3 beschriebenen Festlegung des Busprotokolls und der in 4.4 spezifzierten Bus-Organisation koennen folgende Fehlerklassen auftreten:

Sender:	BF	Bitfehler; Buspegel stimmt nicht mit gesendetem Pegel ueberein.
	BK	Im Zustand TRANSMIT-IDENTIFIER (Arbitrierung) und TRANSMIT-START-OF-FRAME gilt nur Buspegel HIGH bei gesendetem LOW Bit als Fehler
	AF	kein Acknowledge waehrend RECEIVE-ACKNOWLEDGE
	NF	Waehrend des Zustands TEST-INTERMISSION ein LOW Bit auf dem Bus
	SF	Verletzung der Stuffvorschrift
Empfaenger:	CF	CRC-Fehler
	DF	die CRC-SEQUENCE ist nicht mit einem CRC-DELIMITER abgeschlossen; d.h. auf das CRC-Kontrollwort folgt kein HIGH Bit
	NF	Waehrend der Zustaende RECEIVE-END-OF-FRAME und TEST-INTERMISSION ist ein LOW Bit auf dem Bus.
	SF	Verletzung der Stuffvorschrift
	BF	Bitfehler im Zustand TRANSMIT-ACK

In der nachfolgenden Tabelle 4.6.3 sind alle Fehlererkennungsmoeglichkeiten zusammengestellt; die sich aus den Kombinationen aus zeitlichem und raeumlichem Auftreten sowie der Fehlerklassen ergeben. Angegeben sind die zum Zeitpunkt des Fehlers erkennbaren Fehlerklassen.

Die Abkuerzungen in der oberen Zeile jedes Feldes der Matrix charakterisieren die am Sender auftretenden Fehlerklassen; die in der unteren Reihe die an den Empfaengern auftretenden Fehlerklassen.

Tabelle 4.6.3

zeitliches Auftreten	raeumliches	am Sender und allen Empfaeng.	am Sender und einem Teil der Empfaeng	am Sender und kein Empfaenger	nicht am Sender aber allen Empfaenger	nicht am Sender aber Teil der Empfaeng.
BUS-IDLE						
START-OF-FRAME	BK	BK	BK			
IDENTIFIER	BK,SF SF	BK,SF SF	BK,SF	SF	SF	
CONTROL-FIELD (DATA-BYTE-COUNT)	BF,SF SF	BF,SF SF	BF,SF	SF	SF	
DATA-FIELD	BF,SF SF	BF,SF SF	BF,SF	SF	SF	
CRC-SEQUENCE CRC-DELIMITER	BF,SF CF,DF,SF	BF,SF CF,DF,SF	BF,SF	CF,CD,SF	CF,CD,SF	
ACK-SLOT	AF,SF BF,SF	AF,SF BF,SF	AF,SF	BF,SF	BF,SF	
END-OF-BLOCK	BF,NF NF	BF,NF NF	BF,NF	NF	NF	
INTERMISSION	BF,NF NF	BF,NF NF	BF,NF	NF	NF	

4.6.4. Erlaeuterungen zu den Beispielen

In den Bildern werden folgende Abkuerzungen verwendet:

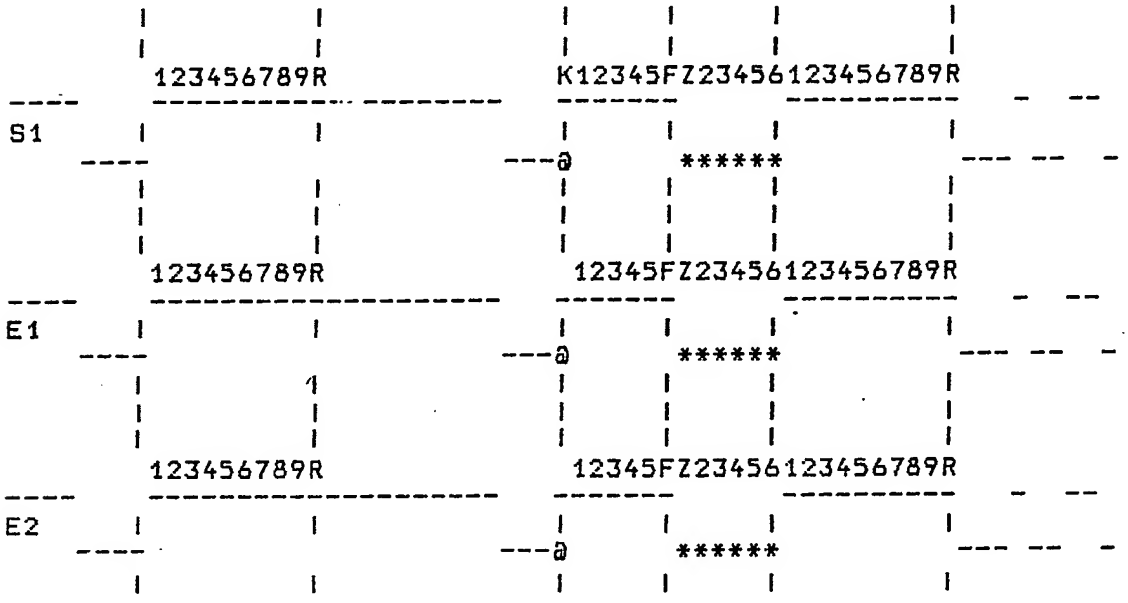
- ∂ Stoerung
- --- ungestoerte Botschaft
- *** eigenes Eingreifen
- ### fremdes Eingreifen
- S Stuffbit
- 1..8. Nummerierung
- R Busruhe (Zustand BUS-IDLE) erkannt
- B Bitfehler waehrend einer Uebertragung erkannt
- F Fehler von einem Teilnehmer erkannt
- Z Beginn einer Fehlermeldung
- K Teilnehmer hat Arbitrierung verloren und bricht seine Uebertragung ab

Modellparameter der folgenden Fehlerbehandlungsbeisiele:

- maximale Anzahl von Bits gleichen Pegels (S = 5)
- END-OF-FRAME besteht aus sechs aufeinanderfolgenden HIGH Bits (EOFL = 6)
- INTERMISSION besteht aus drei aufeinanderfolgenden HIGH Bits (IML = 3)
- eine Fehlermeldung besteht aus mindestens sechs aufeinanderfolgenden LOW Bits (EFL = 6)
- mit Sender wird ein Teilnehmer bezeichnet der entweder schon sendet oder in dem ein Sendewunsch vorliegt

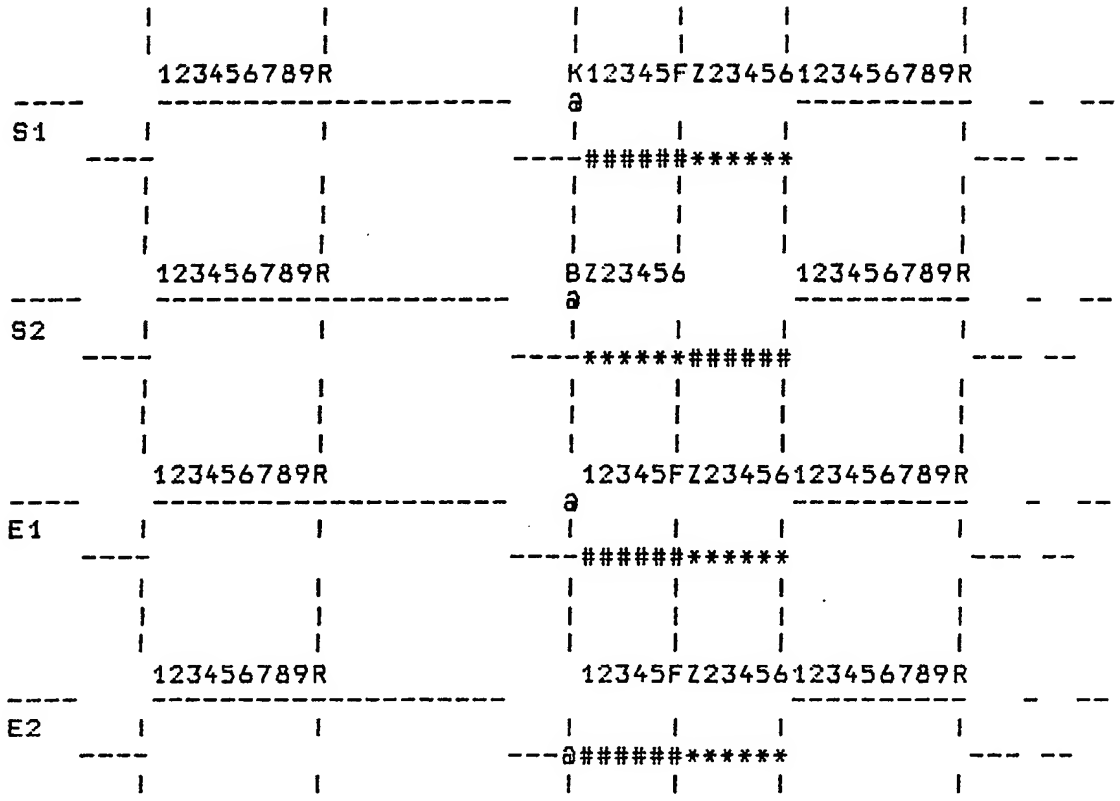
4.6.5. Beispiele Einzelbitfehler

4.6.5.1. Globale Stoerung, von allen Teilnehmern entdeckbar. Bitfehler im IDENTIFIER



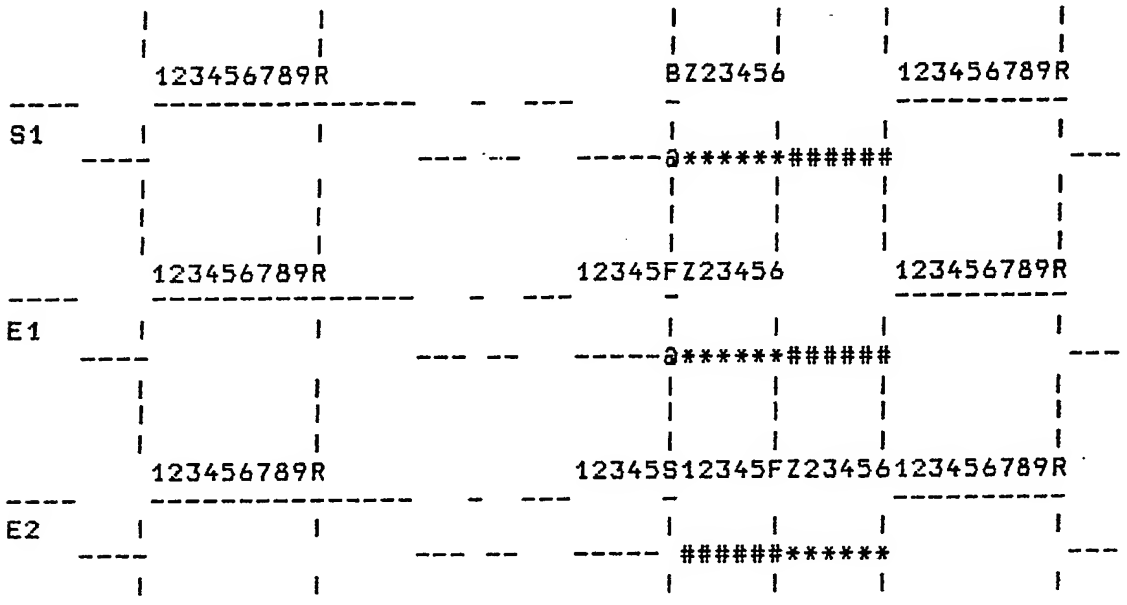
Der Sender 'verliert' die Arbitrierung und erkennt dann, wie die Empfaenger einen Stuffehler. Nach der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachricht begonnen werden.

Zwei Sender beginnen gleichzeitig mit der Uebertragung.



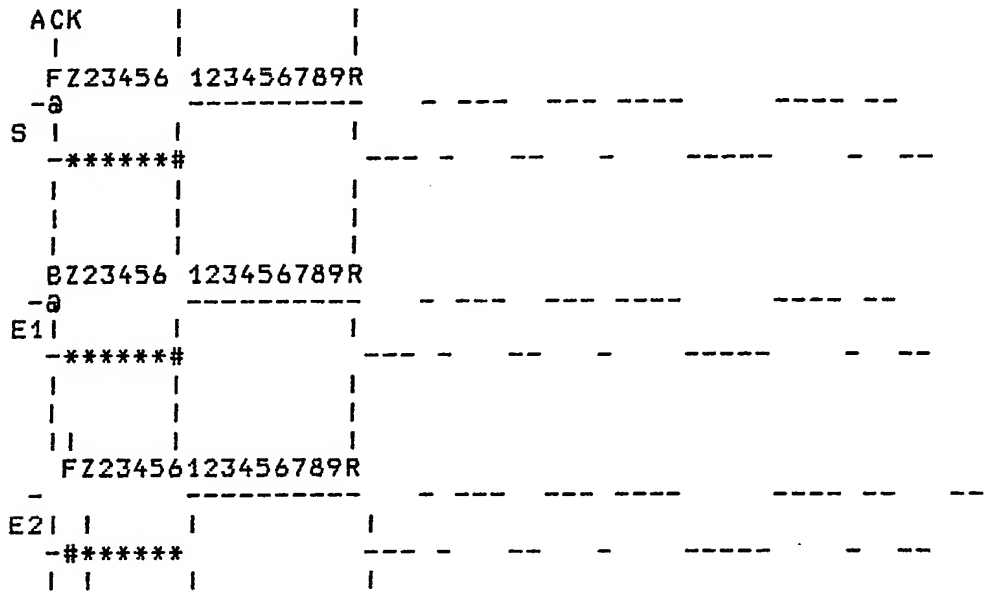
Die Stoerung erzeugt am Sender2 einen Bitfehler, und er setzt seine Fehlermeldung ab. Dadurch kommt es bei den Empfaengern zu einem Stuffehler und am Sender1 entweder zu einem Bitfehler und / oder einem Stuffehler. Nach dem letzten LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachrichten begonnen werden.

4.6.5.2. Fehler tritt an einem Teil der Empfaenger und am Sender bzw. an Station mit Sendewunsch auf. Stuffehler im DATA-FIELD.



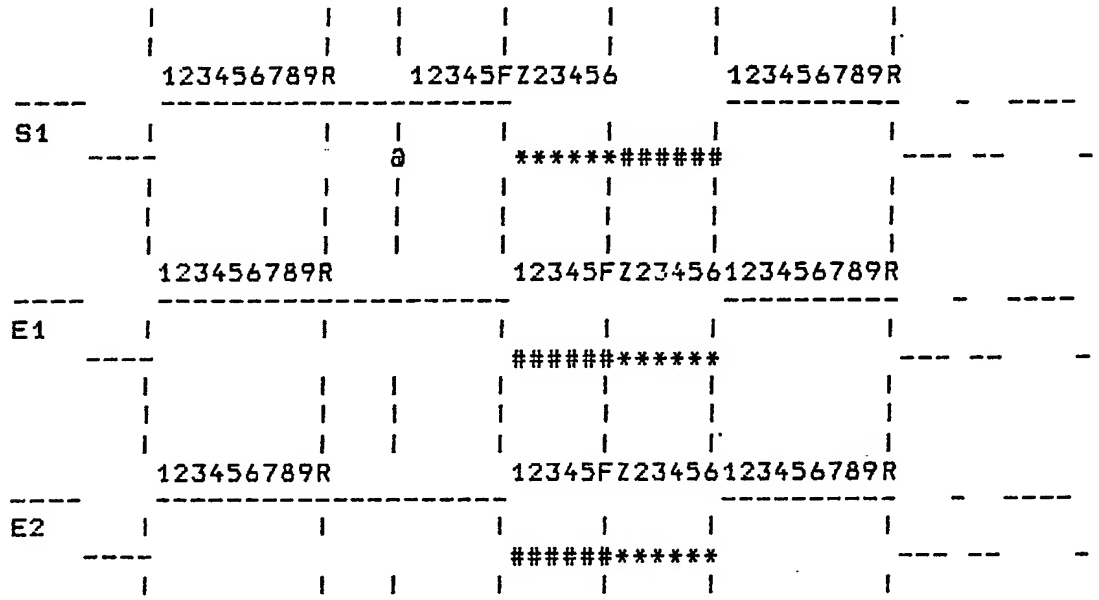
Der Sender erkennt einen Bitfehler und die gestoerten Empfaenger erkennen einen Stuffehler und setzen eine Fehlermeldung ab. Dadurch entsteht an den nicht gestoerten Empfaengern ein Stuffehler und sie setzen ebenfalls eine Fehlermeldung ab. Nach dem letzen LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachricht begonnen werden.

4.6.5.3. Fehler tritt an einem Teil der Empfaenger und am Sender bzw. an Station mit Sendewunsch auf.
Bitfehler im ACK-SLOT



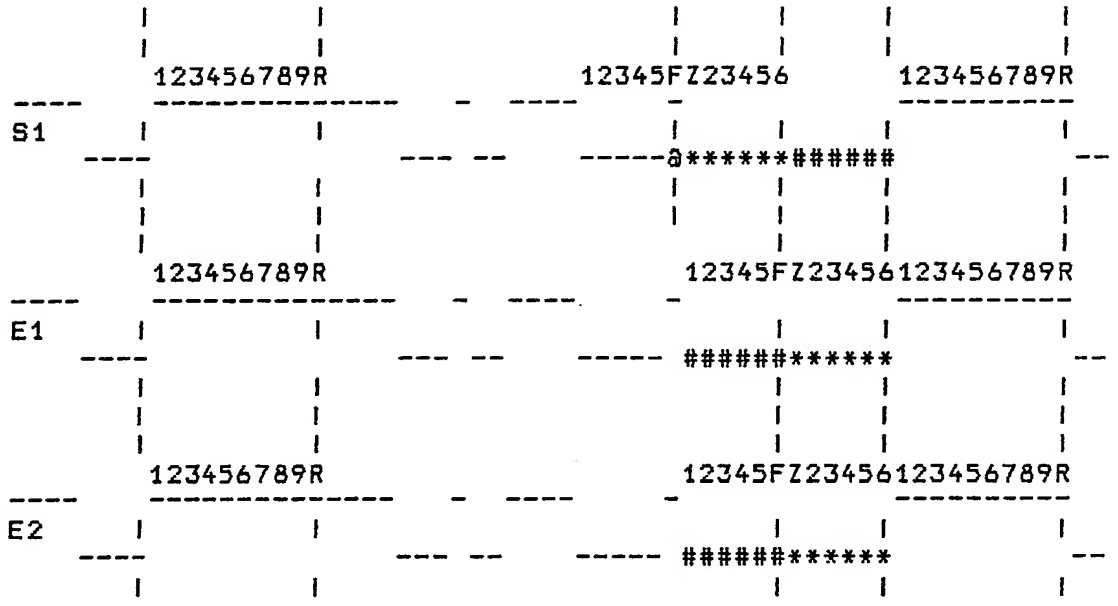
Die gestoerten Empfaenger erkennen einen Bitfehler (ihr LOW Bit wird gestoert) und setzen eine Fehlermeldung ab. Der Sender bekommt kein Acknowledge und setzt ebenfalls eine Fehlermeldung ab. Dadurch erkennen die nicht gestoerten Empfaenger einen Fehler (ACK-DELIMITER) und sie setzten ebenfalls eine Fehlermeldung ab. Nach dem letzten LOW Bit der Fehermeldung kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Botschaft begonnen werden.

4.6.5.4. Nur der Sender wird gestoert. Fehler in BUS-IDLE.



Durch die Stoerung gelingt es dem Teilnehmer S1 nicht mehr seine Uebertragung zu starten, und er verhaelt sich bis zum Ende der Fehlerbehandlung wie ein Empfaenger. S1 erkennt Stuffehler und setzt seine Fehlermeldung ab. Dadurch erhalten die Empfaenger einen Stuffehler und setzen ebenfalls eine Fehlermeldung ab. Nach der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen begonnen werden.

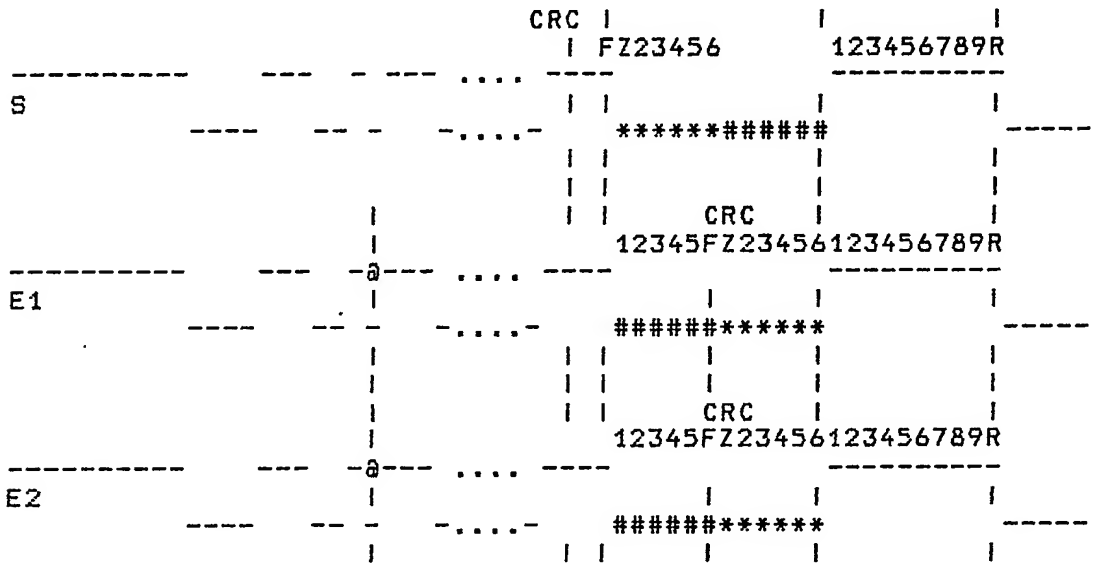
4.6.5.5. Stoerung nur am Sender. Stuffehler im DATA-FIELD



Der Sender erkennt Stuffehler und Bitfehler zugleich und setzt eine Fehlermeldung ab. Dadurch erhalten die Empfaenger einen Stuffehler und setzen ebenfalls eine Fehlermeldung ab. Nach dem letzten LOW Bit der Fehlermeldung und anschliesender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachricht begonnen werden.

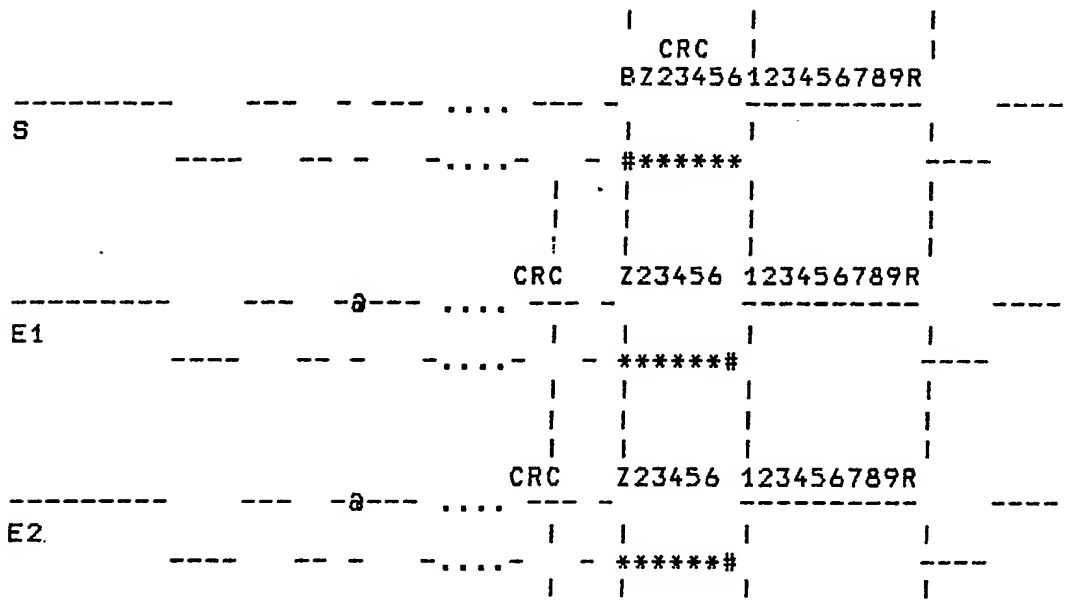
4.6.5.6. Fehler tritt an allen Empfaengern auf; aber nicht am Sender bzw. an Station mit Sendewunsch auf. Bitfehler im CONTROLL-FIELD; Laengenangabe verfaelscht.

Fall 1: Vergroesserung der Laengenangabe; Empfaenger erwarten laengere Botschaft als tatsaechlich gesendet.



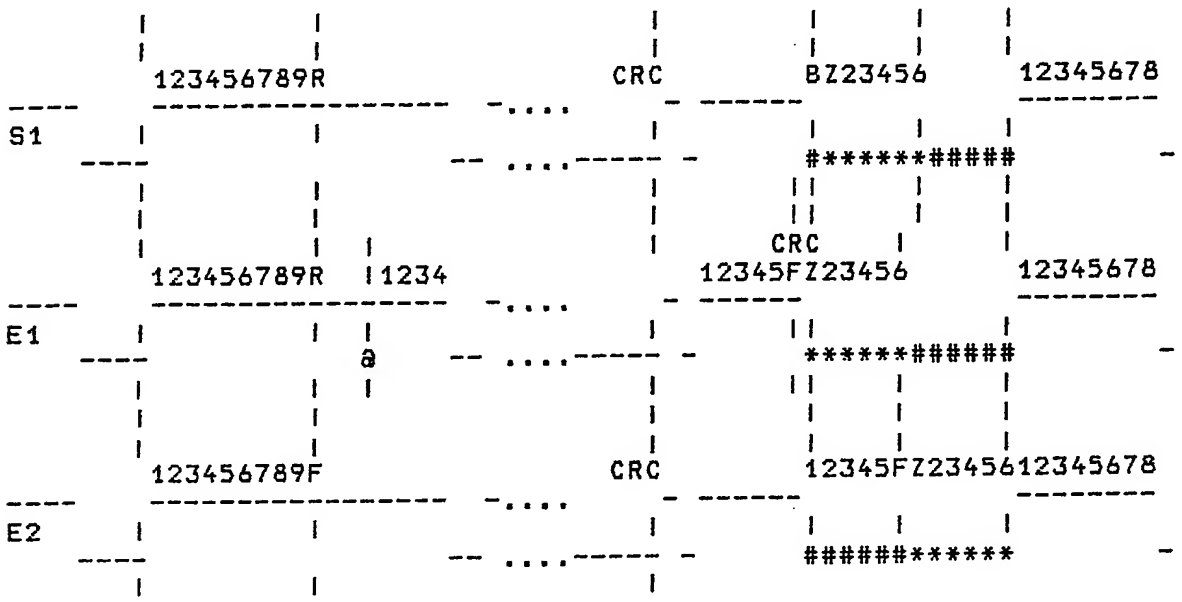
Die Empfaenger koennen nicht mehr an der richtigen Stelle den Empfang einer Botschaft quittieren; der Sender erkennt deswegen einen Fehler und setzt seine Fehlermeldung ab. Dadurch entsteht an den Empfaengern ein Stuffehler und sie setzen ebenfalls eine Fehlermeldung ab. Nach dem letzen LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Botschaft begonnen werden.

Fall 2: Verkleinerung der Laengenangabe;
Empfaenger erwarten eine kuerzere Botschaft als tatsaechlich gesendet.



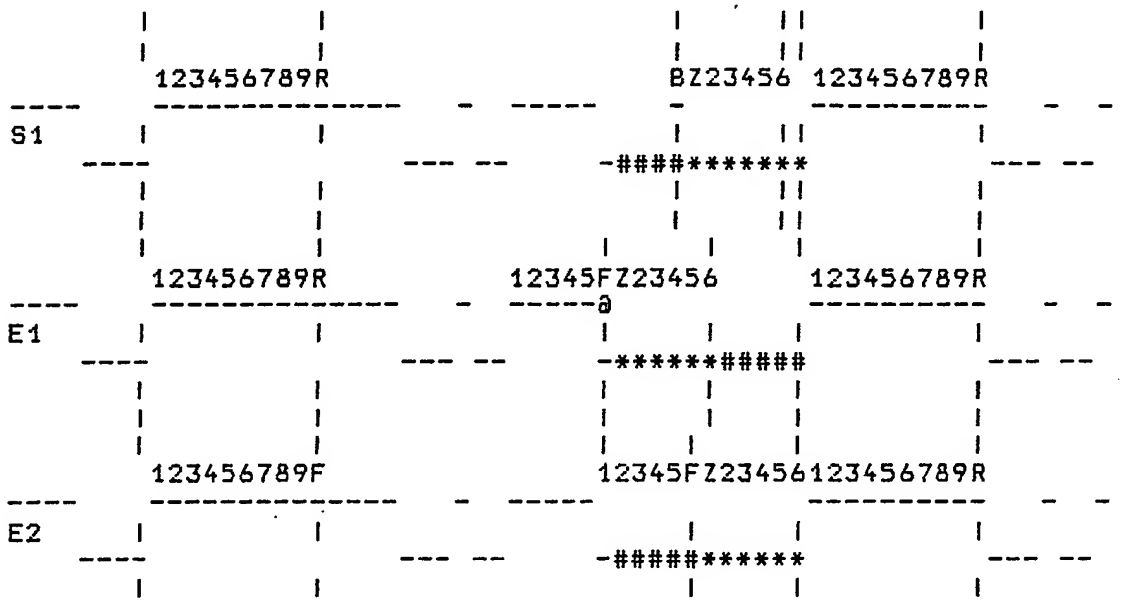
Die Empfaenger stellen einen CRC-Fehler fest und setzen eine Fehlermeldung ab. Dadurch entsteht am Sender ein Bitfehler und er setzt ebenfalls eine Fehlermeldung ab. Nach dem letzen LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Botschaft begonnen werden.

4.6.5.7. Fehler tritt an einem Teil der Empfaeger auf; aber nicht am Sender bzw. an Station mit Sendewunsch. Bitfehler in BUS-IDLE.



Die gestoerten Empfaenger erhalten durch die Stoerung eine falsche Laengenangabe und setzen deshalb ihren CRC-Check an die falsche Stelle. Wenn der CRC der gestoerten Empfaenger um mehr als 8 Takte spaeter als die richtige Lage erfolgt, so erhalten die Empfaenger einen Stuffehler und setzen eine Fehlermeldung ab. Erfolgt der CRC-Check innerhalb von 8 Takten nach der richtigen Lage, so erkennen die gestoerten Empfaenger einen CRC-Fehler und senden eine Fehlermeldung. Dadurch entsteht am Sender ein Bitfehler. Die nicht gestoerten Empfaenger empfangen als END-OF-FRAME eine unzulaessige Bitfolge und setzen ebenfalls eine Fehlermeldung ab. Nach dem letzten LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachricht begonnen werden.

4.6.5.8. Fehler tritt an einem Teil der Empfaeger auf; aber nicht am Sender bzw. an Station mit Sendewunsch. Stuffehler in DATA-FIELD.



Die gestoerten Empfaenger erkennen einen Stuffehler und setzen eine Fehlermeldung ab. Dadurch entsteht am Sender ein Bitfehler und an den nicht gestoerten Empfaengern ein Stuffehler und sie setzen ebenfalls eine Fehlermeldung ab. Nach dem letzten LOW Bit der Fehlermeldung und anschliessender Busruhe (Zustand BUS-IDLE) kann mit neuen Uebertragungen oder der Wiederholung der gestoerten Nachricht begonnen werden.

4.7. INTERFACE-MANAGEMENT-PROCESSOR (IMP)

4.7.1. Konfiguration

4.7.1.1. Allgemeines Konzept

4.7.1.1.1. Struktur

Der IMP hat folgende Aufgaben

- Datenaustausch zwischen CPU und serieller Schnittstelle durch die Benutzung eines DUAL PORT RAM (DPRAM)
- Steuerung von Senden und Empfangen.

Fig. 11 zeigt das Blockschaltbild des seriellen Schnittstellenbausteins Der serielle Schnittstellenbaustein besteht aus den Teilen :

- DPRAM
- IMP
- serielles Schieberegister
- Buslogik
- Taktoszilator

Die Verbindungen der Teile untereinander sind im Blockschaltbild angegeben.

4.7.1.1.2. Prioritaet der Botschaften

Die Prioritaet innerhalb des IMP wird durch den Platz der verschiedenen Botschaftem im DPRAM festgelegt. Die Prioritaet auf dem Bus (Arbitrierung) wird durch den IDENTIFIER der Botschaft bestimmt.
Der Suchprozess nach der hoechstprioren Botschaft innerhalb eines IMP startet an der niedrigsten DPRAM Adresse.

4.7.1.1.3. Akzeptanzfilter

Jede vom Bus ankommende Nachricht wird daraufhin geprueft, ob sie empfangen werden soll oder nicht. Dazu wird im DPRAM die Liste durchgesehen, in der alle Botschaften stehen, die in diesem IMP verarbeitet werden. Eine ankommende Botschaft wird nur dann ins DPRAM uebernommen, wenn ihr IDENTIFIER dort gefunden wird und wenn sie auch empfangen werden soll, was durch das RECEIVE/TRANSMIT Bit anzeigt wird.

4.7.1.1.4. Warteschlange der Botschaften, die gesendet werden soll.

Jede Botschaft, die gesendet werden soll, wird von der CPU in das DPRAM geschrieben, wobei anschliessend das TRANSMIT-REQUEST Bit gesetzt wird. Ein Suchvorgang findet die hoechstpriore Botschaft, die zur Uebertragung ansteht. Nur diese Botschaft wird fuer eine Uebertragung ins serielle Schieberegister beruecksichtigt. Wenn die CPU spaeter eine wichtigere Botschaft zur Uebertragung anmeldet, so wird der Suchvorgang diese entdecken und die erste Botschaft verdraengen. Dadurch werden die Botschaften entsprechend ihrer Prioritaet uebertragen, unabhaenig von ihrer Ankunftszeit.

4.7.1.1.5. Warteschlange empfangenen Botschaften

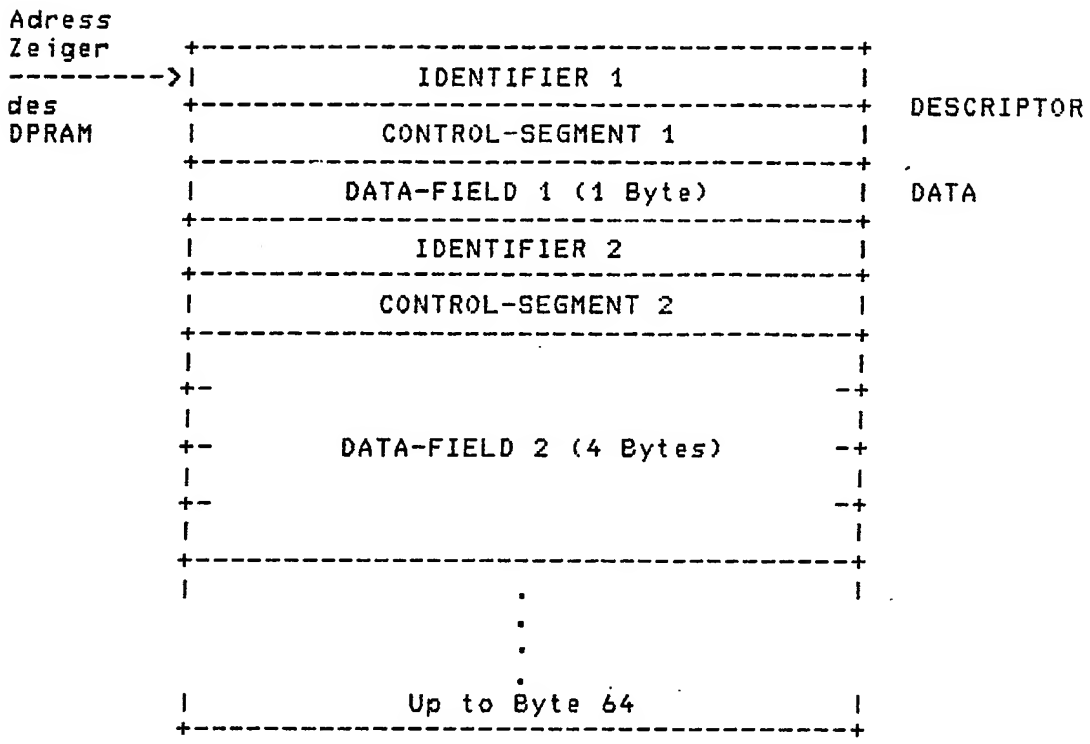
Wenn empfangene Botschaften im DPRAM gespeichert werden, wird automatisch das INTERRUPT-REQUEST Bit gesetzt. Der Suchvorgang wird unter den empfangenen Botschaften diejenige mit der hoechsten Prioritaet finden und einen Interrupt an die CPU weitergeben . Wenn eine wichtigere Botschaft im DPRAM gepseichert wird, bevor die CPU den Interrupt beantwortet hat, so wird die hoeherpriore Botschaft beruecksichtigt. Damit werden die Botschaften entsprechend ihrer Prioritaet beruecksichtigt, unabhaengig von ihrer Ankunftszeit.

4.7.1.2. DPRAM Organisation

Das DPRAM enthaelt DESCRIPTORen (IDENTIFIER, CONTROL-SEGMENTe) und DATA-FIELDs aller Botschaften, die von der CPU ueber den Bus empfangen oder gesendet werden sollen. Die Aufteilung des DPRAM ist unabhaengig vom spezifizierten Protokoll. Moeglichkeiten:

- getrennte Speicher fuer ankommende und abgehende Botschaften
- getrennte Speicher fuer DESCRIPTORs und DATA-FIELDS
- einen einzigen Speicher fuer alle Aufgaben

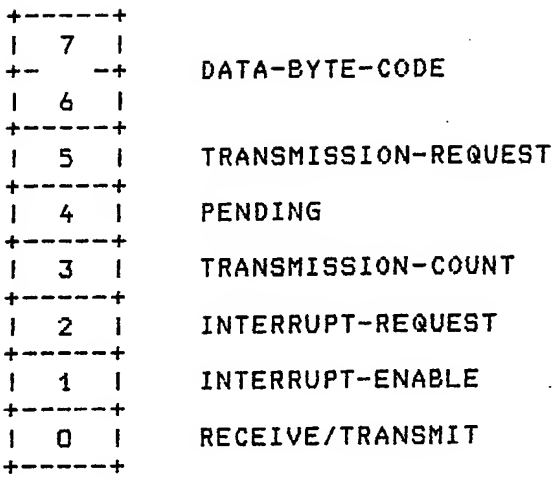
In einer ersten Ausfuehrung wird ein Speicher fuer alle Aufgaben vorgeschlagen. Die Botschaften koennen aequidistant in den Speicher eingetragen werden. Um jedoch Speicherplatz zu sparen, ist es angebracht, die Botschaften hintereinander dichtgepackt abzuspeichern, mit einer Laenge von drei bis zehn Bytes abhaengig von der DATA-BYTE-COUNT.
Die Groesse des DPRAM ist zur Zeit auf 64 Byte festgelegt. Wenn die Integragtionskosten in Zukunft kleiner werden sollten, kann der Speicher vergroessert werden. Damit koennten eine groessere Anzahl von Botschaften oder laengere Botschaften (sowohl DATA-FIELD als auch DESCRIPTOR) gespeichert werden. Es koennten auch Ersatzbotschaften fuer das Ausbleiben von Nachrichten im DPRAM abgelegt werden, was im Fehlerfall die Software vereinfachen wuerde.



Das DPRAM dient also als:

- Speicher, der gleichzeitig entscheidet ob ankommende Botschaften empfangen werden sollen (Akzeptanzfilter)
- Warteschlange fuer ankommende und fortgehende Botschaften geordnet nach ihrer Prioritaet
- Speicher fuer die Kontrollregister des IMP

4.7.1.3. CONTROL-SEGMENT Organisation



4.7.1.4. Funktion des DPRAM

4.7.1.4.1. DATA-BYTE-CODE

Bei Suchvorgaengen muss der Adress Zeiger des DPRAM zuerst auf den jeweiligen Anfang einer Botschaft zeigen. Wenn die DESCRIPTORen und DATA-Fields dichtgepackt abgespeichert sind, kann auf die naechste botschaft gezeigt werden indem man den Zeiger um

2**(DATA-BYTE-COUNT) + 2

erhoeht. Um den Speicherplatz und die Botschaften besser auszunuetzen, koennen mehrere Botschaften unter einem DESCRIPTOR zusammengefasst werden und als eine grosse Botschaft uebertragen werden. Die maximale Blockgroesse wird vorerst auf 8 Bytes festgelegt.
Die Laenge des DATA-FIELDs erhaelt man aus dem DATA-BYTE-CODE mit

DATA-BYTE-COUNT = 2** DATA-BYTE-CODE

4.7.1.4.2. PENDING

Das PENDING Bit zeigt an, dass entweder eine Uebertragung oder eine Interruptroutine noch nicht fertig ist. Es wird automatisch gesetzt wenn eine Uebertragung auf dem Bus beginnt oder wenn die CPU einen Interrupt bestaetigt (Laden des Interruptzeigers). Das PENDING Bit wird automatisch nach einer Uebertragung (erfolgreich oder nicht erfolgreich) zurueckgesetzt. Das PENDING Bit wird ausserdem unter Programmkontrolle von der CPU zusammen mit dem INTERRUPT-REQUEST oder bei Ankunft einer neuen Botschaft mit dem gleichen IDENTIFIER zurueckgesetzt. Ist das PENDING Bit einer Botschaft gesetzt, so wird diese Botschaft bei einem Suchvorgang nicht beruecksichtigt.

Das PENDING Bit erlaubt dem Programmierer der CPU am Ende der Interruptroutine zu ueberpruefen, ob ein weiterer Interrupt die Konsistenz einer Botschaft mit mehreren Bytes zerstoert hat.*** Wenn das PENDING Bit am Ende der Interruptroutine immer noch auf HIGH steht ,ist dies die Bestaetigung, dass das DATA-FIELD konsistent bearbeitet wurde vor Ankunft der naechsten Botschaft.

INTERRUPT-REQUEST	PENDING	Wirkung
LOW	LOW	kein Interrupt
LOW	HIGH	kein Interrupt
HIGH	LOW	neuer Interrupt
HIGH	HIGH	Bestaetigung des Interrupts

Diese Vorgaenge erfolgen nur fuer INTERRUPT-ENABLE = HIGH. Wenn der INERRUPT-ENABLE auf Low zurueckgesetzt ist, dann wird das PENDING Bit nicht mit der Interruptbestaetigung der CPU gesetzt.

4.7.1.4.3. INTERRUPT-ENABLE

Ein gesetztes INTERRUPT-ENABLE erlaubt, dass INTERRUPT-REQUESTs im richtigen Bezug zur CPU weitergegeben werden.
Ein nicht gesetztes INTERRUPT-ENABLE verhindert, dass eine Interruptaktion an die CPU gemeldet wird.

4.7.1.4.4. INTERRUPT-REQUEST

Der INTERRUPT-REQUEST wird automatisch auf HIGH gesetzt bei jeder Uebertragung einer neu empfangenen Botschaft ins DPRAM unabhaengig von Zustand des INTERRUPT-ENABLE. INTERRUPT-REQUEST wird auch gesetzt, wenn eine wiederholte Uebertragung (TRANSMISSION-COUNT = HIGH) scheitert.

4.7.1.4.5. TRANSMISSION-COUNT

TRANSMISSION-COUNT zaehlt die Ubertragungsversuche. TRANSMISSION-COUNT wird nach einer erfolgreichen Uebertragung zurueckgesetzt.

TRANSMISSION-REQUEST	TRANSMISSION-COUNT	Zustand
LOW	LOW	keine Uebertragung
LOW	HIGH	keine Uebertragung
HIGH	LOW	erste Uebertragung
HIGH	HIGH	zweite Uebertragung

Nach der zweiten fehlerhaften Uebertragung wird INTERRUPT-REQUEST gesetzt um die weitere Fehlerbehandlung der Benutzersoftware ueberlassen zu koennen.

4.7.1.4.6. TRANSMISSION-REQUEST

Ein gesetztes TRANSMISSION-REQUEST Bit veranlasst den IMP die zugehoerige Botschaft zu uebertragen.
Es gibt zwei verschiedene Zustaende:

a. RECEIVE/TRANSMIT = LOW (Uebertragung)
   Startwert nach einem Reset. Die Botschaft wird gesendet. TRANSMISSION-REQUEST wird von der CPU oder von einer ankommenden Botschaft mit gesetzem REMOTE-TRANSMISSION-REQUEST Bit auf HIGH gesetzt.

b. RECEIVE/TRANSMIT = HIGH (Empfang)
   Alle so gekennzeichneten Botschaften sind im Empfangsmodus. Als Ausnahme in diesem Zustand wird durchOP Setzen von TRANSMISSION-REQUEST eine Botschaft mit leerem DATA-FIELD und gesetztem REMOTE-TRANSMISSION-REQUEST abgeschickt.

4.7.1.4.7. RECEIVE/TRANSMIT

RECEIVE/TRANSMIT bestimmt die Richtung der Daten.

a. RECEIVE/TRANSMIT = LOW (Uebertragung)
   Startwert nach einem Reset. Die DATA-FIELDs im DPRAM sind fuer ankommende Botschaften schreibgeschuetzt, wenn RECEIVE/TRANSMIT auf LOW gesetzt ist. Die Botschaften werden durch TRANSMISSION-REQUEST HIGH aktiviert.

   Es gibt jedoch eine Ausnahme:
   Wenn REMOTE-TRANSMISSION-REQUEST = HIGH in einer ankommenden Botschaft ist, dann wird TRANSMISSION-REQUEST des zugehoerigen CONTROL-SEGMENTs auf HIGH gesetzt trotz RECEIVE/TRANSMIT = LOW. TRANSMISSION-REQUEST ist das einzige Bit, das in dieser Operation geaendert wird, alle anderen Bits bleiben weiterhin schreibgeschuetzt. Dieses Vorgehen dient zur Anforderung von Botschaften durch andere Busteilnehmer.
b. RECEIVE/TRANSMIT = HIGH (Empfang)
   Die ankommenden Botschaften werden in das zugehoerige DATA-FIELD uebertragen. INTERRUPT-REQUEST wird automatisch auf HIGH gesetzt bei jeder Uebertragung der zugehoerigen Botschaft.

4.7.2. Datenaustausch CPU-DPRAM

4.7.2.1. Synchronisations Problem

Der Austausch von DATA-FIELDs, die aus mehreren Bytes bestehen, kann abhaengig von der Anwendung ohne Beruecksichtigung von der synchronen Betriebweise vorgenommen werden. Wenn die DATA-FIELDs jedoch synchron verarbeitet werden sollen, muss eine geeignete Synchronisation vorgesehen werden. Um die Hardwarekosten niedrig zu halten, wird zur Zeit eine Software-Synchronisation vorgeschlagen. In dieser Vorgehenweise greift die CPU auf das DPRAM im Cycle Stealing mit Prioritaet gegenueber allen anderen Zugriffsversuchen zu.

4.7.2.2. Nicht synchrone Operation

Das DPRAM wird von der CPU als RAM-Erweiterung benutzt. Es werden keine weiteren Uebertragungsbefehle dazu benoetigt. Empfangene Botschaften werden einfach in das DPRAM geschrieben, wobei die vorhergehenden Botschaften ueberschrieben werden. Uebertragungen werden von der CPU durch Setzen von TRANSMISSION-REQUEST im CONTROL-SEGMENT, das zu dem zu uebertragenen DATA-FIELD gehoert, gestartet. TRANSMISSION-REQUEST kann ebenso von fremden CPUs durch REMOTE-TRANSMISSION-REQUEST gesetzt werden.

4.7.2.3. Software Synchronisation

4.7.2.3.1. Abfragen einer Sequenznummer

Wenn die sendende CPU ein Byte des DATA-FIELDs als Sequenznummer benutzt, und diese vor jedem Setzen des TRANSMISSION-REQUEST Bits inkrementiert, dann kann die empfangende CPU anhand der Sequenznummer pruefen, ob sich diese veraendert hat nachdem sie die Daten im DATA-FIELD bearbeitet hat. Wenn sich die Sequenznummer veraendert hat, muss die empfangende CPU einen Teil der Bearbeitung mit den neuen Daten wiederholen.

4.7.2.3.2. Abfragen des INTERRUPT-REQUEST

Neu ankommende Botschaften setzen automatisch das INTERRUPT-REQUEST Bit im zugehoerigen CONTROL-SEGMENT. Auch wenn der CPU wegen INTERRUPT-ENABLE = LOW keinen Interrupt mitgeteilt wird, wird INTERRUPT-REQUEST vor jeder Bearbeitung der Botschaft zuruechgesetzt und nach der Bearbeitung wird geprueft, ob es nicht in der Zwischenzeit wieder gesetzt wurde.

4.7.2.3.3. Interrupt gesteuerte Bearbeitung

Die empfangende CPU kann im CONTROL-SEGMENT der zu empfangenden Botschaft das INTERRUPT-ENABLE Bit setzen. Neu ankommende Botschaften setzen das zugehoerige INTERRUPT-REQUEST Bit und setzen gleichzeitig automatisch das PENDING Bit auf LOW zurueck. Der wichtigste Interrupt wird der CPU gemeldet. Die Bestaetigung der CPU setzt automatisch das PENDING Bit auf HIGH. Bevor die CPU am Ende der Interruptbehandlung beide (INTERRUPT-REQUEST und PENDING) zurueecksetzt, fragt sie das PENDING Bit ab. Wenn das PENDING Bit immer noch auf HIGH ist, so ist die naechste Botschaft nicht innnerhalb der Interruptbehandlung angekommen.

4.7.2.3.4. Block Uebertragung

Um DATA-FIELDs synchron zu bearbeiten koennen diese blockweise zu und von CPU uebertragen werden. Ausreichender RAM Speicherplatz muss fuer die dadurch bedingte doppelte Speicherung bereitgestellt werden.

a. Senden
Am Anfang wird TRANSMISSION-REQUEST, das als Semaphor Variable dient, geprueft. Wenn es zurueckgesetzt ist, wird das DATA-FIELD byteweise vom CPU-RAM ins DPRAM uebertragen. Danach wird TRANSMISSION-REQUEST gesetzt.

WARNUNG: Bei dieser Vorgehensweise kann im Falle eines gesetzten REMOTE-TRANSMISSION-REQUEST Bits keine korrekte Synchronisation garantiert werden.

b. Empfang
Bevor die eigentliche Interruptroutine begonnen wird , wird das DATA-FIELD blockweise ins CPU-RAM uebertragen, wie unter 4.7.2.4.2. beschrieben. Dadurch wird die Wahrscheinlichkeit, dass die Daten inkonsistent werden, betraechtlich verringert, da der kritische Zeitabschnitt von der gesamten Interruptbearbeitung auf die Blockuebertragung des DATA-FIELDs verringert wird.

4.7.2.3.5. Doppelter Empfangspuffer

Fuer wichtige Botschaften koennen zwei Speicherplaetze im DPRAM vorgesehen werden. Ein IDENTIFIER erscheint dann zweimal im DPRAM. Die Benutzersoftware kann nun ankommende Botschaften von einem Speicherplatz im DPRAM auf einen anderen durch Aendern des zugehoerigen RECEIVE/TRANSMIT Bits, anschliessend an die Bestaetigung des Interrupts, umschalten. Bei dieser Vorgehensweise ist in einem Speicherplatz das RECEIVE/TRANSMIT Bit HIGH, und ist damit bereit, die entsprechende Botschaft aufzunehmen, und im anderen Speicherplatz mit dem gleichen IDENTIFIER ist das RECEIVE/TRANSMIT Bit LOW, und damit ist die gerade empfangene Botschaft dort schreibgeschuetzt.

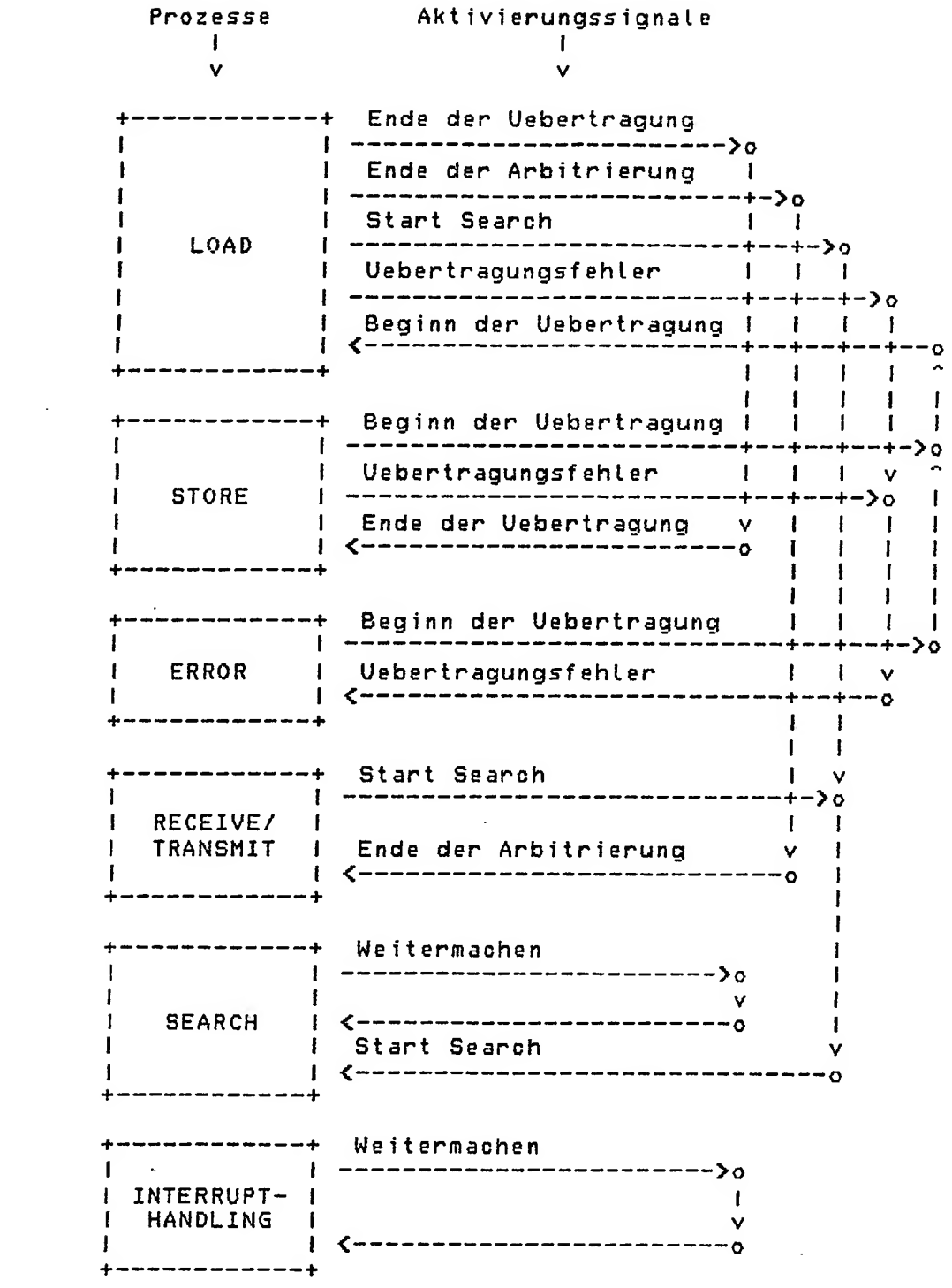
4.7.2.4. Hardware Synchronisation

Als Voraussetzung fuer eine Hardware Synchronisation muss die CPU schnelle Blockuebertragung mit DMA-Faehigkeiten und nicht unterbrechbare Semaphorbehandlung bieten. Der Zugriff auf das DPRAM wird durch wechselseitigen, durch ein Semaphorbit geregelten Zugriff von der CPU und dem IMP vor inkonsistenter Blockuebertragung geschuetzt. Zwischen dem Ende einer Uebertragung und dem Start einer neuen muss genuegend Zeit vorgesehen werden. Im schlechtesten Fall muss der IMP solange warten, bis die CPU eine Blockuebertragung abgeschlossen hat. Die CPU muss im schlechtesten Fall warten, bis der IMP eine empfangene Botschaft im DPRAM gespeichert und die naechste Botschaft, die uebertragen werden soll, ins serielle Schieberegister geladen hat.

4.7.3. Datenaustausch DPRAM - Schieberegister

4.7.3.1. Parallele Steuerprozesse

Der Datenaustausch zwischen DPRAM und seriellem Schieberegister wird durch mehrere parallele Prozesse gesteuert, deren Funktion spaeter beschrieben wird. Ein Prozess wird initialisiert und damit in den aktiven Zustand ueberfuehrt durch ein Aktivierungssignal. Im aktiven Zustand kann jeder Prozess weitere Aktivierungssignale ausgeben, die wiederum zu anderen Prozessen gehen. Die Ausgabe eines Aktivierungssignals bedeutet nicht notwendigerweise, dass der Quellprozess sich gleichzeitig deaktivieren muss. Die gesamte Steuerung enthaelt die folgenden Prozesse und Aktivierungssignale:



Aktivierungssignale:

- Ende der Uebertragung:
  Endesignal am Ende einer fehlerfrei abgeschlossenen Uebertragung. Wird beim Uebergang aus dem Zustand TRANSMIT-END-OF-FRAME in den Zustand TEST-INTERMISSION fuer einen Bustakt gesetzt. - Ende der Arbitrierung:
  Signal am Ende des Arbitrierungsvorgangs; nach Auftreten dieses Signals ist der BUS-Zugriff geregelt (Senden oder Empfangen). Wird am Ende der Uebertragung des IDENTIFIERs fuer einen Bustakt gesetzt.

- Start Search:
  Startsignal fuer den SEARCH-Prozess, vom Anfang des DPRAM an den Suchvorgang neu zu beginnen; erfolgt am Ende des RECEIVE/TRANSMIT-Prozesses.

- Uebertragungsfehler:
  Fehlermeldung vom Uebertragungsvorgang. Wird fuer einen Bustakt gesetzt, wenn eine Fehlermeldung auf dem Bus uebertragen wird.

- Beginn der Uebertragung:
  Startsignal fuer den LOAD-Prozess, erfolgt am Ende des STORE-Prozesses oder des ERROR-Prozesses

- Weitermachen:
  - Erneuter Start bei Endlos-Prozessen

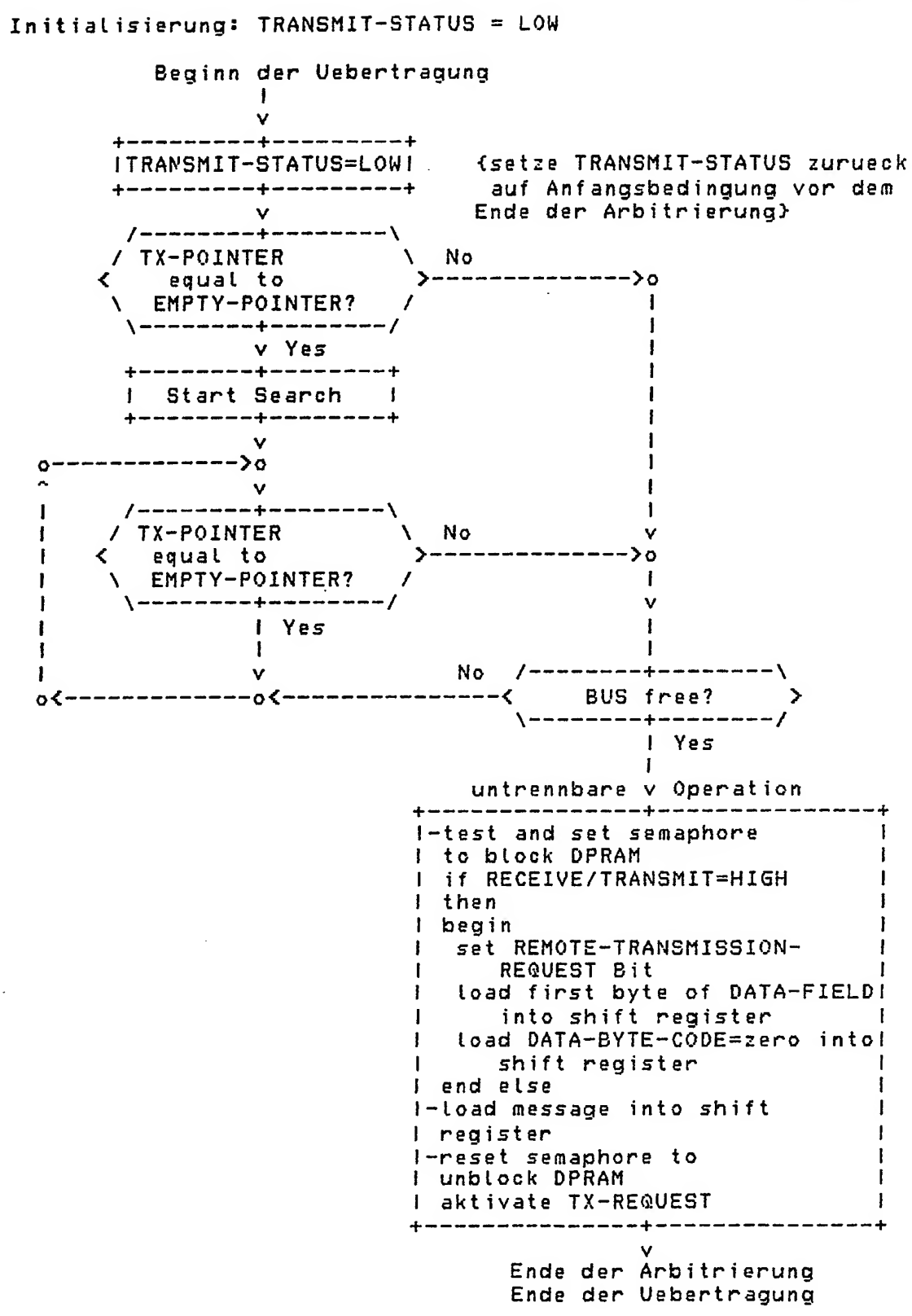
Die Prozesse sind im folgenden unter Zuhilfenahme der ueblichen Kontrollfluss-Konstrukte erklaert, wie sie in ALGOL oder PASCAL verwendet werden und mit denen auch in der Informatik-Literatur gearbeitet wird.

4.7.3.2. LOAD-Prozess

Der LOAD-Prozess laedt die naechste zu uebertragende Botschaft in das serielle Schieberegister. Wenn keine Botschaft zur Uebertragung ansteht, wird der SEARCH-Prozess gestartet. Wenn der SEARCH-Prozess bei noch belegtem BUS eventuell eine zweite Botschaft mit hoeherer Prioritaet in der Warteschlange zu uebertragender Prozesse findet, so wird diese Botschaft die vorher gefundene verdraengen.

Bus free:
  Bus free ist wahr, waehrend dIE BOTSCHAFTSSEGMENTE START-OF-FRAME ,IDENTIFIER, CONTROL-FIELD, DATA-FIELD, CRC-FIELD UND ACK-FIELD uebertragen werden.

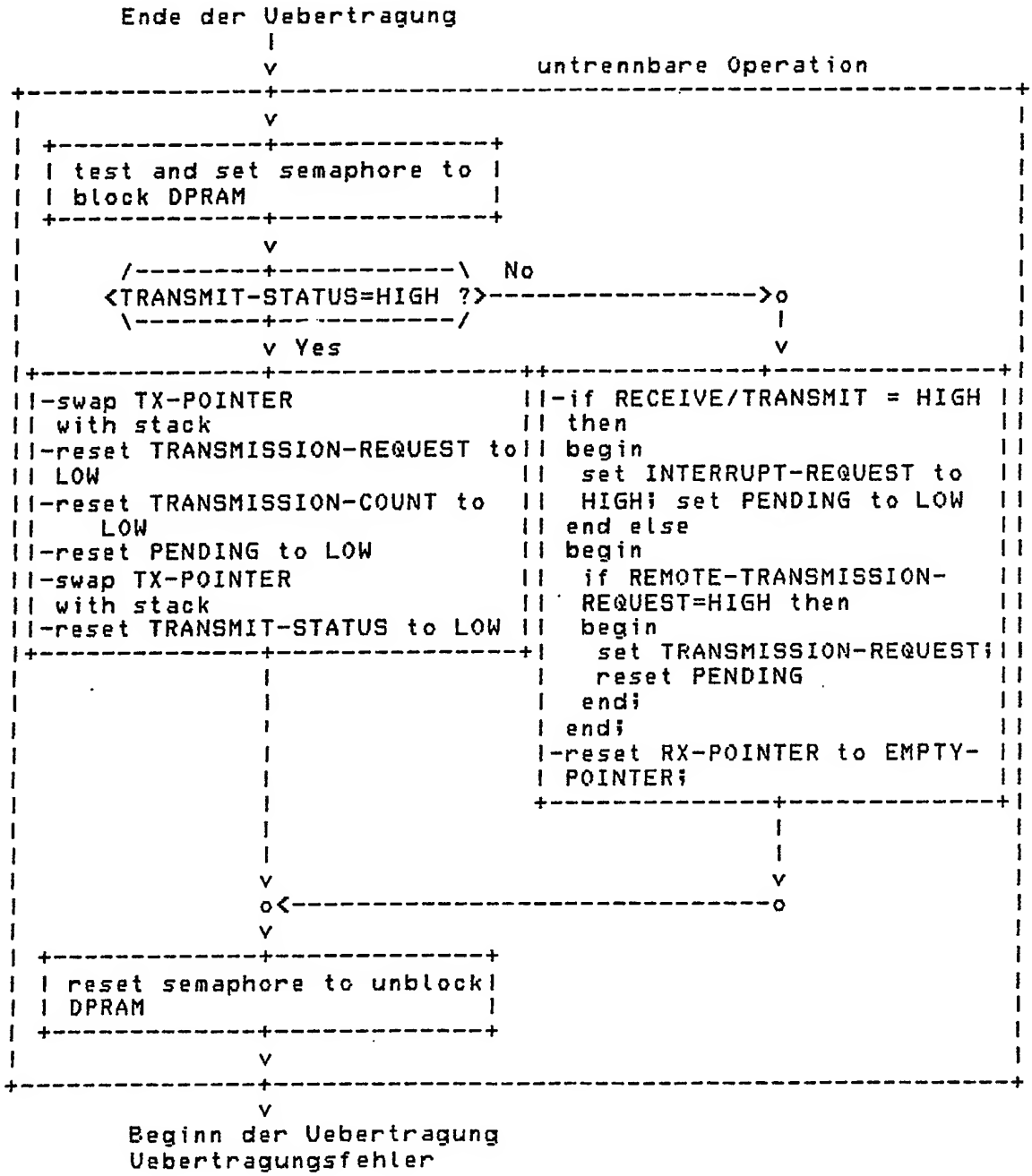
TRANSMIT-STATUS:
  Muss am Ende der Arbitierung gesetzt (HIGH) oder rueckgesetzt (LOW) werden


Initialisierung: TRANSMIT-STATUS = LOW
Beginn der Uebertragung
TRANSMIT-STATUS=LOW
{setze TRANSMIT-STATUS zurueck auf Anfangsbedingung vor dem Ende der Arbitrierung}
TX-POINTER equal to EMPTY-POINTER?
No
Yes
Start Search
TX-POINTER equal to EMPTY-POINTER?
No
Yes
BUS free?
No
Yes
untrennbare Operation
-test and set semaphore to block DPRAM
if RECEIVE/TRANSMIT=HIGH
then
begin
set REMOTE-TRANSMISSION-REQUEST Bit
load first byte of DATA-FIELD into shift register
load DATA-BYTE-CODE=zero into shift register
end else
-load message into shift register
-reset semaphore to unblock DPRAM
aktivate TX-REQUEST
Ende der Arbitrierung
Ende der Uebertragung

Uebertragungsfehler

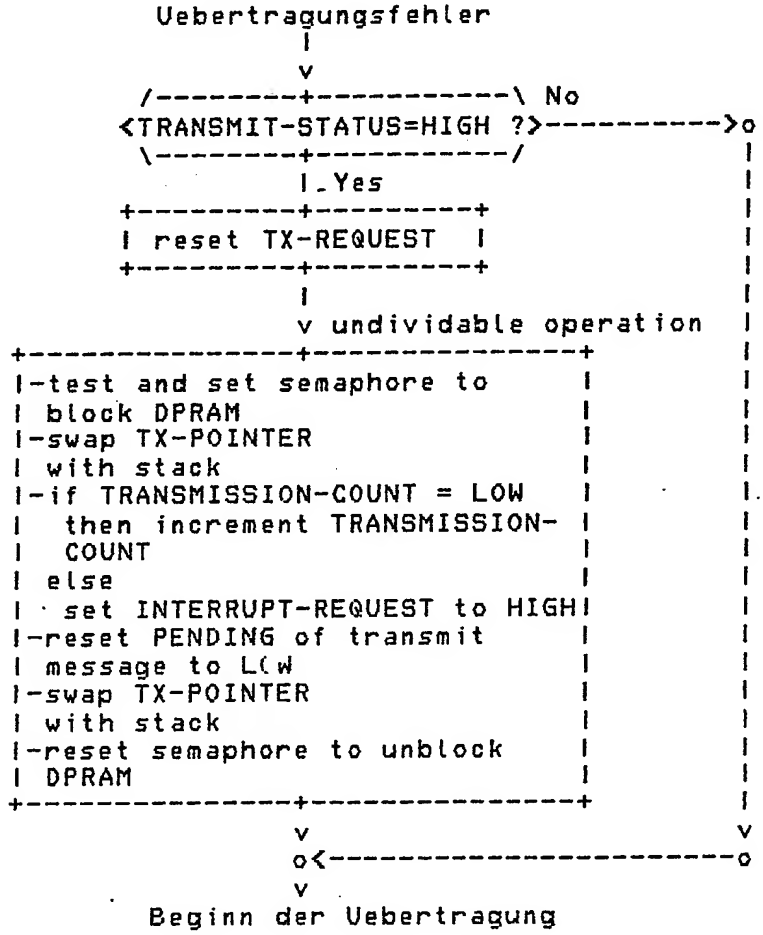
4.7.3.3. STORE-Prozess

Der STORE-Prozess speichert entweder eine empfangene Botschaft oder verwaltet die TRANSMISSION-REQUEST und PENDING Bits von gesendeten Botschaften. Das Signal Ende der Uebertragung kommt nur nach einem erfolgreichen Uebertragungsvorgang ohne Fehler. Im Falle eines Fehlers kommt Uebertragungsfehler.


Ende der Uebertragung
untrennbare Operation
test and set semaphore to block DPRAM
<TRANSMIT-STATUS=HIGH ?>
No
Yes
-swap TX-POINTER with stack
-reset TRANSMISSION-REQUEST to LOW
-reset TRANSMISSION-COUNT to LOW
-reset PENDING to LOW
-swap TX-POINTER with stack
-reset TRANSMIT-STATUS to LOW
-if RECEIVE/TRANSMIT = HIGH then begin set INTERRUPT-REQUEST to HIGH; set PENDING to LOW end else begin if REMOTE-TRANSMISSION-REQUEST=HIGH then begin set TRANSMISSION-REQUEST; reset PENDING end; end;
-reset RX-POINTER to EMPTY-POINTER;
reset semaphore to unblock DPRAM
Beginn der Uebertragung
Uebertragungsfehler

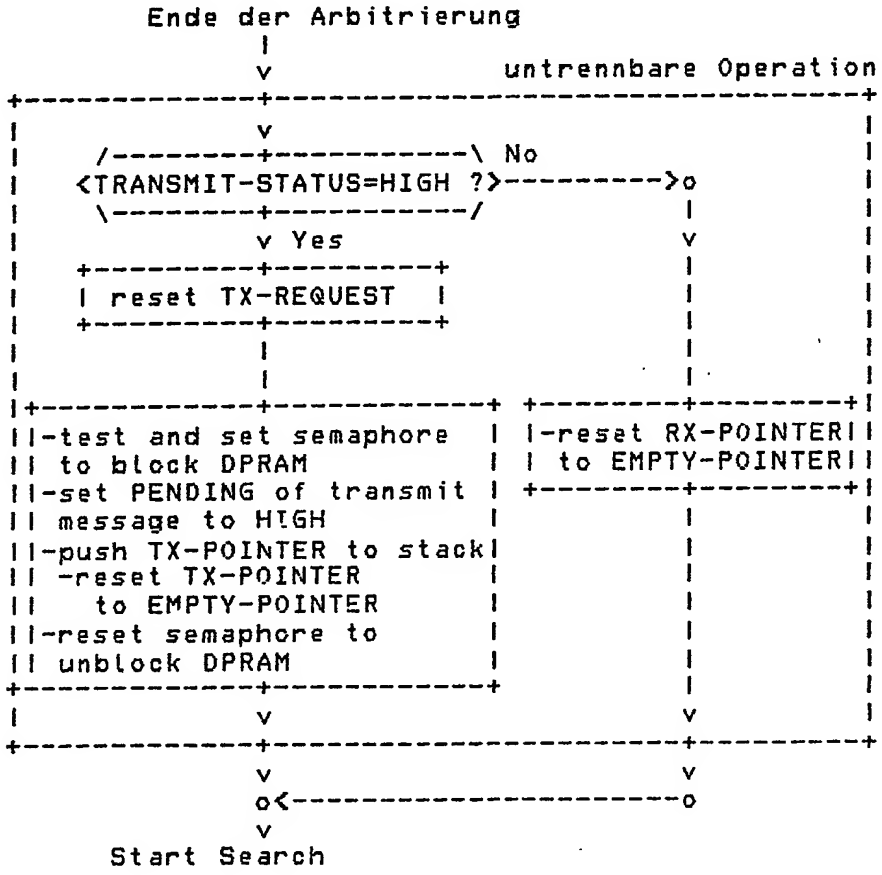
4.7.3.4. ERROR-Prozess

Der Error Prozess zaehlt im Fehlerfall TRANSMISSION-COUNT hoch oder setzt im Falle einer zweiten nicht gegluckten Uebertragung INTERRUPT-REQUEST. Eine empfangene Botschaft wird im Fehlerfall einfach nicht weiter verarbeitet.



4.7.3.5. RECEIVE/TRANSMIT-Prozess

Im Falle dass die Arbitrierung gewonnen wurde, wird das PENDING Bit, das zur gerade uebertragenen Botschaft gehoert, auf HIGH gesetzt und dann der zugehoerige TX-POINTER auf den Stack geladen. Der RECEIVE/ TRANSMIT-Prozess setzt je nach Ausgang der Arbitrierung den TX-POINTER oder den RX-POINTER auf einen leeren Adresszeiger zurueck, bevor der SEARCH-Prozess durch Start Search aktiviert wird.

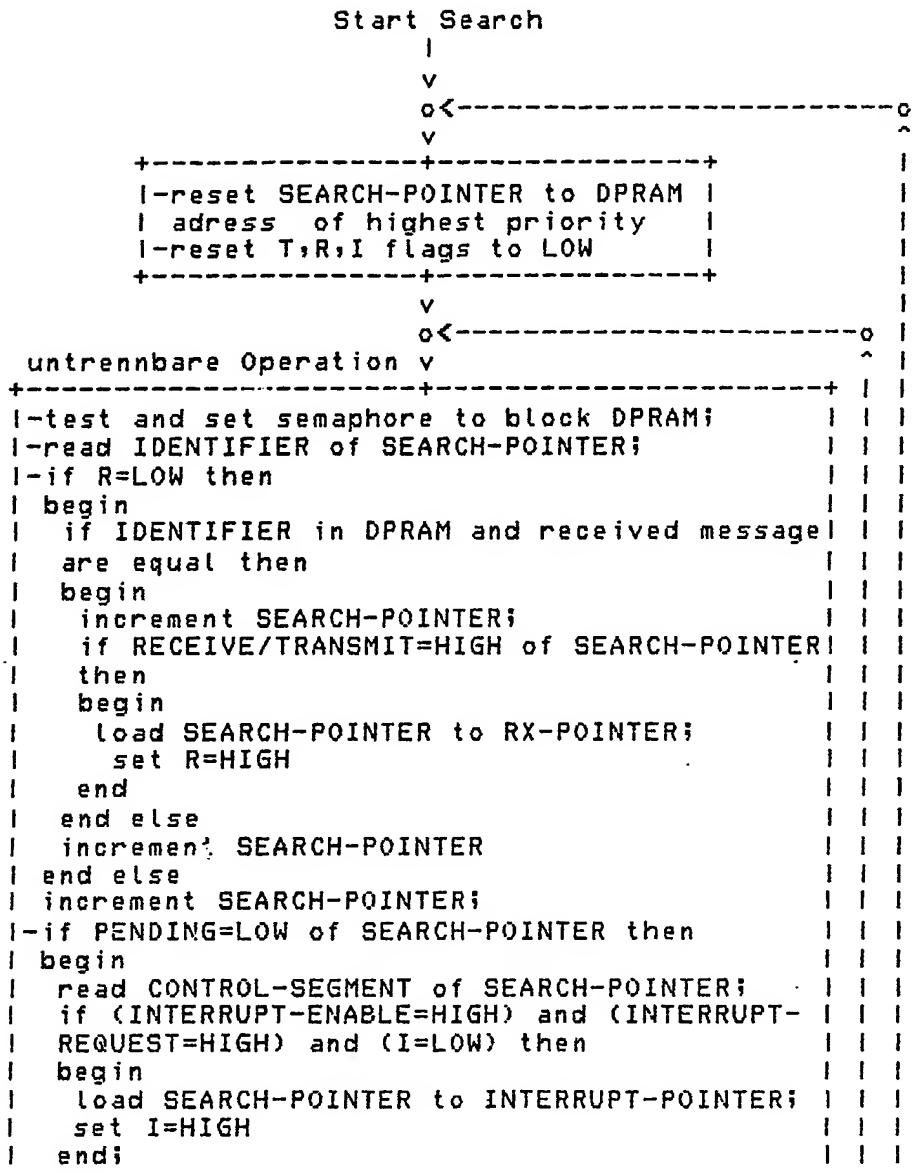


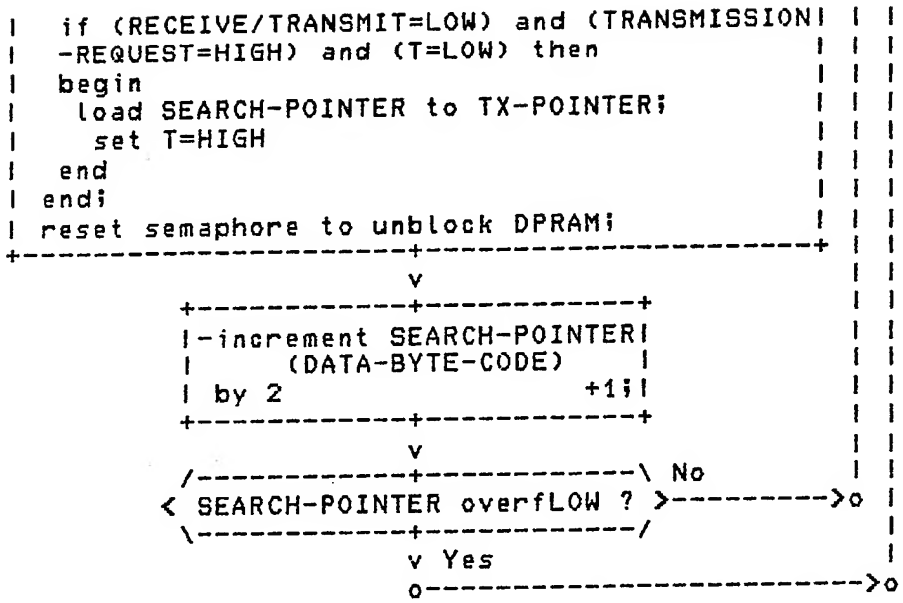
4.7.3.6. SEARCH-Prozess

Der Search-Prozess sucht andauernd nach Adresszeigern des DPRAM fuer

- zu uebertragende Botschaften (TX-POINTER), wenn TRANSMISSION-REQUEST gesetzt ist,
- zu empfangende Botschaften (RX-POINTER), dh. ob der empfangene IDENTIFIER im DPRAM enthalten ist,
- Botschaften, die eine Unterbrechungsanforderung bei der CPU ausloesen sollen, dh. bei denen INTERRUPT-REQUEST und INTERRUPT-ENABLE gesetzt sind.

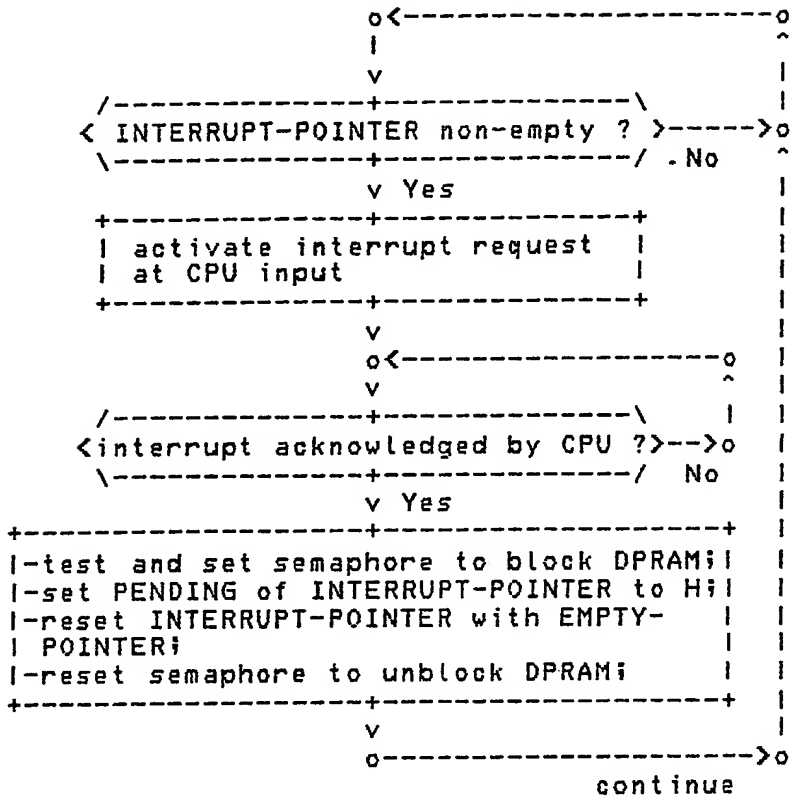
Der SEARCH-Prozess wird mit Start Search auf die Botschaft mit der hoechsten Prioritaet am Anfang des DPRAM zurueckgesetzt, wenn die Echtzeitablaeufe bei der Uebertragung auf dem BUS es erfordern. Das Absuchen einer gesamten Botschaft ist als unteilbare Operation ausgefuehrt. In jeder Clock-Periode kann der SEARCH-Prozess durch einen DPRAM-Zugriff der CPU im Cycle-Stealing-Verfahren angehalten und um eine Clock-Periode verzoegert werden. Nach Abschluss der unteilbaren Operation kann der SEARCH-Prozess durch hoeherpriore Prozesse vom Zugriff des DPRAM ausgeschlossen werden, bis die Semaphore-Variable wieder vom SEARCH-Prozess selbst gesetzt werden kann.


Start Search
-reset SEARCH-POINTER to DPRAM adress of highest priority
-reset T;R;I flags to LOW
untrennbare Operation
-test and set semaphore to block DPRAM;
-read IDENTIFIER of SEARCH-POINTER;
-if R=LOW then
begin
if IDENTIFIER in DPRAM and received message are equal then
begin
increment SEARCH-POINTER;
if RECEIVE/TRANSMIT=HIGH of SEARCH-POINTER then
begin
load SEARCH-POINTER to RX-POINTER;
set R=HIGH
end
end else
increment SEARCH-POINTER
end else
increment SEARCH-POINTER;
-if PENDING=LOW of SEARCH-POINTER then
begin
read CONTROL-SEGMENT of SEARCH-POINTER;
if (INTERRUPT-ENABLE=HIGH) and (INTERRUPT-REQUEST=HIGH) and (I=LOW) then
begin
load SEARCH-POINTER to INTERRUPT-POINTER;
set I=HIGH
end;



4.7.3.7. INTERRUPT-HANDLING-Prozess

Der INTERRUPT-HANDLING-Prozess leitet Interrupts von empfangenen Bo schaften oder von mehrfach fehlerhaft uebertragenen Botschaften an die CPU weiter. Es wird jeweils der Interrupt mit der durch die Anordnung im DPRAM vorgegebenen hoechsten Prioritaet an die CPU weitergeleitet. Der INTERRUPT-HANDLING-Prozess laeuft ununterbrochen, parallel zu den aenderen Prozessen. Der Anfangswert des Interrupt Pointers ist der leere Adresszeiger.



4.7.4. Steuerung des Zugriffs zum DPRAM

4.7.4.1. Zugriffssynchronisation

Die Zugriffe zum DPRAM werden durch eine Semaphore-Variable geregelt. Diese sichert den unteilbaren Zugriff zu den zusammengehoerigen Bytes der DATA-FIELDs und des CONTROL-SEGMENTs. Andere Zugriffe werden solange blockiert, bis die Semaphore- Variable zurueckgesetzt ist. Damit wird die Konsistenz von DATA-FIELDs geschuetzt, die laenger als ein Byte sind. Wenn mehrere Prozesse versuchen sollten, die Semaphore-Variable gleichzeitig zu setzen, dann gilt das folgende Prioritaeten- Schema:

a. STORE
b. LOAD
c. ERROR
d. INTERRUPT-HANDLING
e. RECEIVE/TRANSMIT

f. SEARCH

Die CPU wird von der Zugriffsverwaltung mittels Semaphore ausgenommen. Sie hat per Cycle-Stealing immer sofortigen Vorrang vor allen anderen Zugriffen. Es ist klar, dass mit dieser Regelung die Konsistenz der Daten bei CPU-Zugriffen ggfs. verletzt werden kann. Um dies zu vermeiden, muesste bekanntlich auch die CPU in die Zugriffsregelung mit Semaphore einbezogen werden. Dies erforderte aber spezielle Befehle zur unteilbaren Behandlung von Semaphores, die bei den derzeit am Markt verfuegbaren CPUs noch nicht realisiert sind.

4.7.4.2. DPRAM-Adresszeiger

Auf den DPRAM wird von verschiedenen Quellen aus zugegriffen, der CPU und den parallelen Prozessen. Die Adressen kommen dabei von

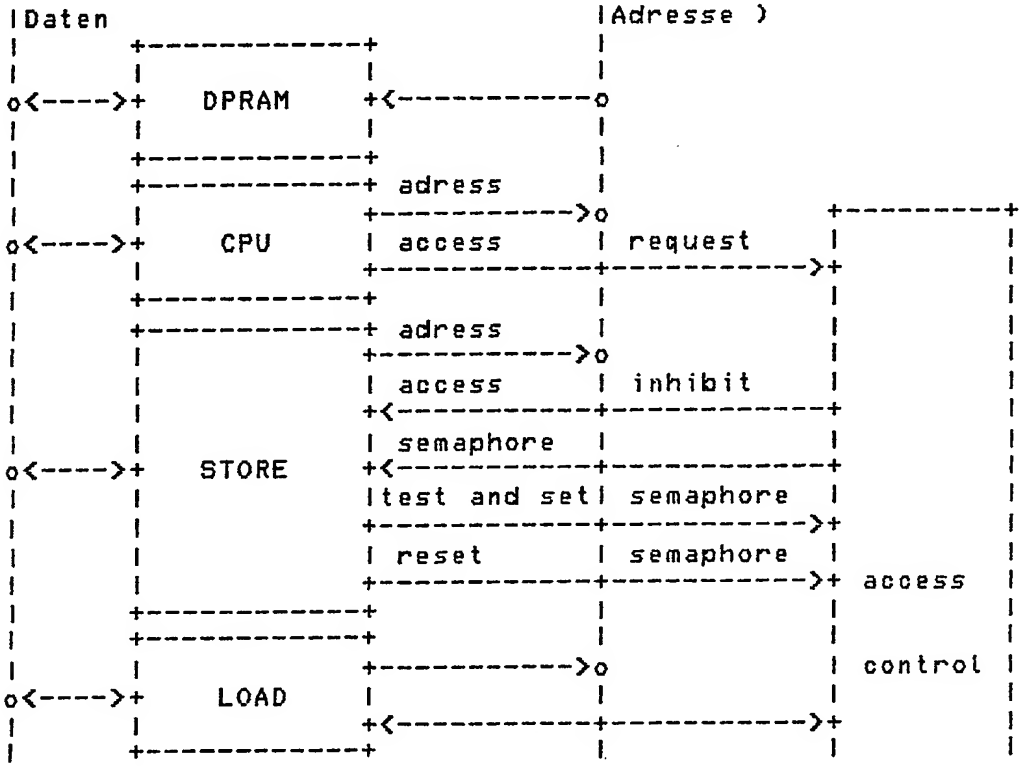
- CPU-Adresse
- EMPTY-POINTER:
  Adresszeiger, der auf eine leere, d.h. nicht mit sinnvollen Botschaften gefuellte Adresse zeigt. Wenn ein Adresszeiger gleich dem EMPTY-POINTER ist, wird dies so interpretiert, dass keine entsprechende Botschaft vorliegt
- SEARCH-POINTER:
  Adresszeiger des SEARCH-Prozesses
- TX-POINTER:
  Adresszeiger, der auf die zu sendende Botschaft zeigt. Falls keine Botschaft zu senden ist, ist der TX-POINTER = EMPTY-POINTER
- RX-POINTER:
  Adresszeiger, der auf die zu empfangende Botschaft zeigt. Falls keine Botschaft zu empfangen ist, ist der RX-POINTER = EMPTY-POINTER
- INTERRUPT-POINTER:
  Adresszeiger, der auf die Botschaft zeigt, die eine Unterbrechungsanforderung an die CPU hat. Falls keine Unterbrechungsanforderung vorliegt, ist der INTERRUPT-POINTER = EMPTY-POINTER

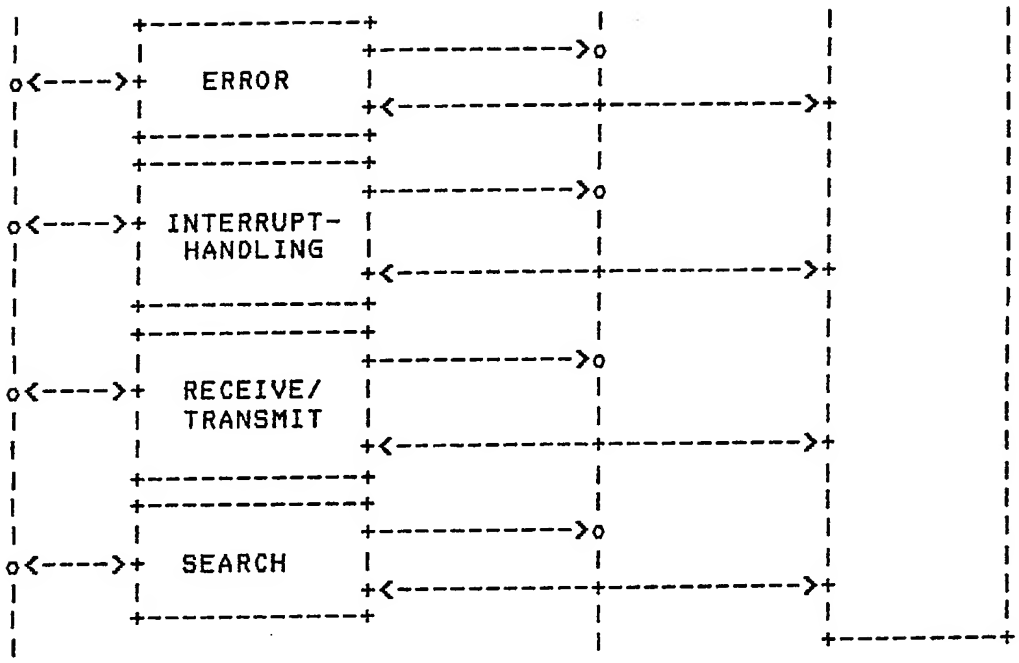
Die Adresszeiger (Pointer) werden in den verschiedenen Prozessen verarbeitet, um die DPRAM-Adresse zu erhalten, unter der Daten gespeichert oder gelesen werden sollen. Die tatsaechlichen DPRAM- Zugriffe mittels der Adresszeiger sind entsprechend ihrer Prioritaet organisiert. Unteilbare Zugriffe

auf mehrere Bytes geschehen mittels der Semaphore-Variablen; die den Zugriff fuer alle Prozesse blockiert; ausser fuer den Prozess, der die Semaphore-Variable selbst gesetzt hat.

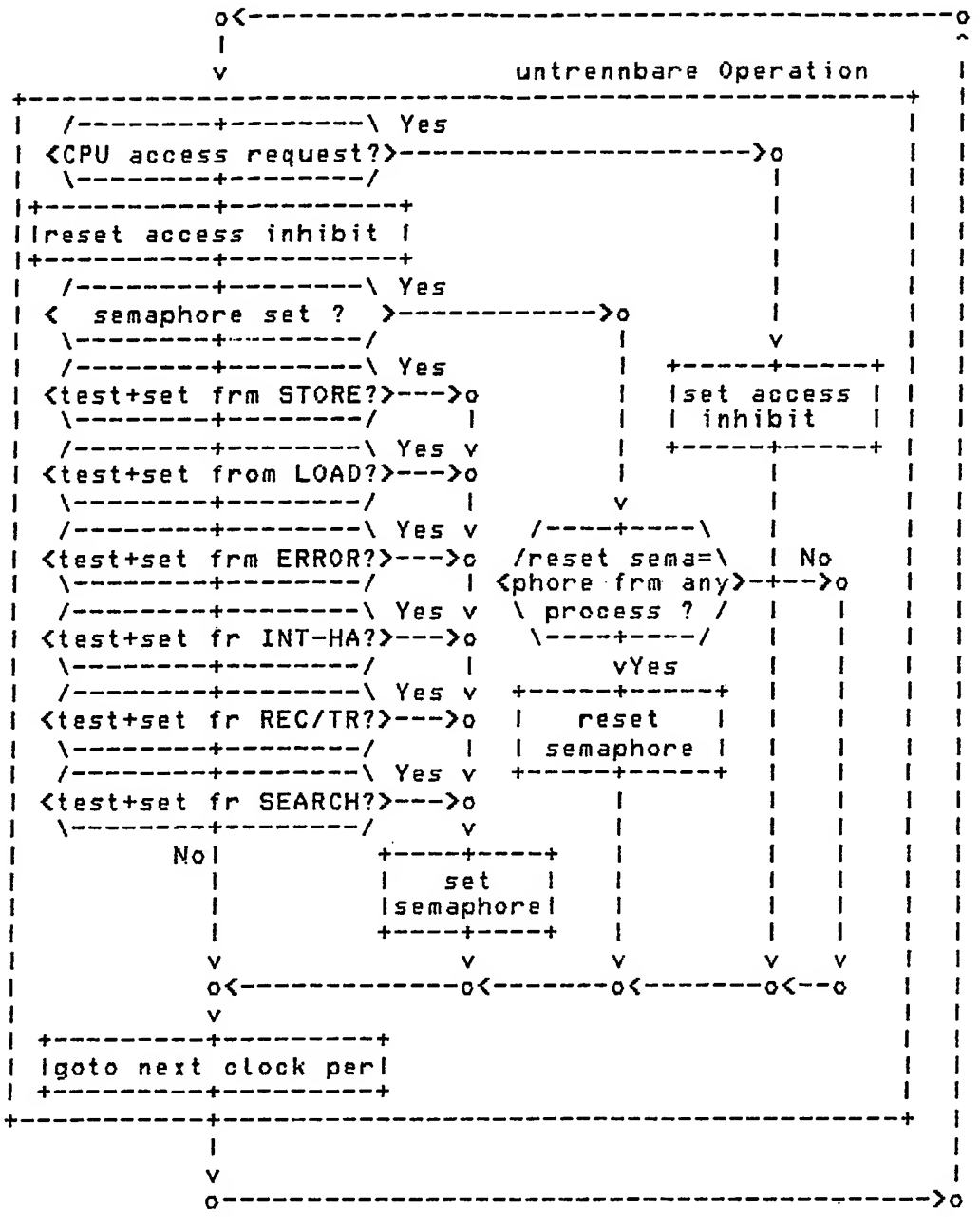
4.7.4.3. Prioritaetssteuerung

Wie bereits beschrieben, werden der CPU-Zugriff auf den DPRAM und das Setzen der Semaphore-Variablen nach Prioritaeten geordnet. Ein Prozess darf seine Adresse nur dann an den Adresseingang des DPRAM legen, wenn die CPU nicht zugreifen will (kein Access request und deshalb auch kein Access inhibit) und wenn die Semaphore-Variable nicht von einem anderen Prozess belegt ist (Semaphore = LOW). Die Anschluesse von LOAD, ERROR, INTERRUPT-HANDLING, RECEIVE/TRANSMIT und SEARCH sind gleich wie bei STORE und deshalb nicht detailliert gezeichnet.





Eine Ausfuehrung fuer den Block 'Access Control' des obigen Schemas ist im folgenden Flussdiagramm angegeben, das in jeder Clock-Periode durchlaufen wird.



Im obigen Flussdiagramm wurden die Abkuerzungen verwendet:
- INT-HA for INTERRUPT-HANDLING
- REC/TR for RECEIVE/TRANSMIT and
- per for period.

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

SEITE

INHALTSVERZEICHNIS

INHALTSVERZEICHNIS

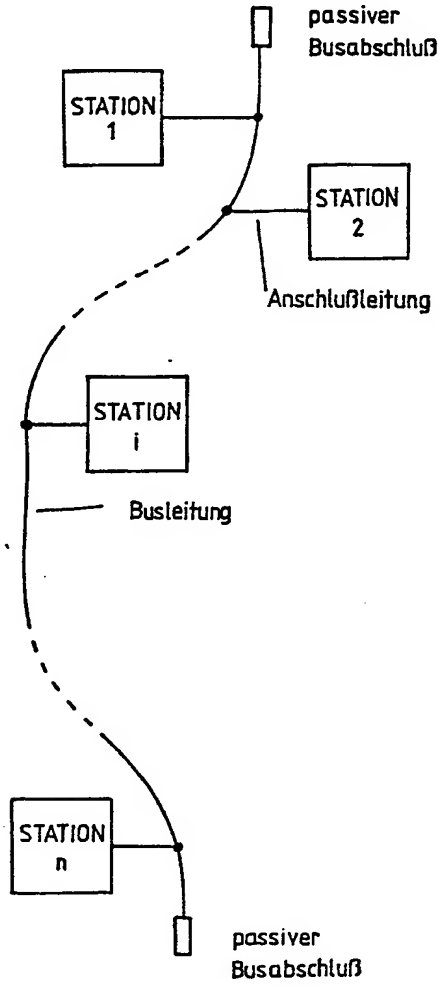
SEITE

– Leerseite –

Numm r: 35 06 118
Int. Cl.[4]: G 06 F 15/16
Anmeld tag: 22. Februar 1985
Offenl gungstag: 28. August 1986

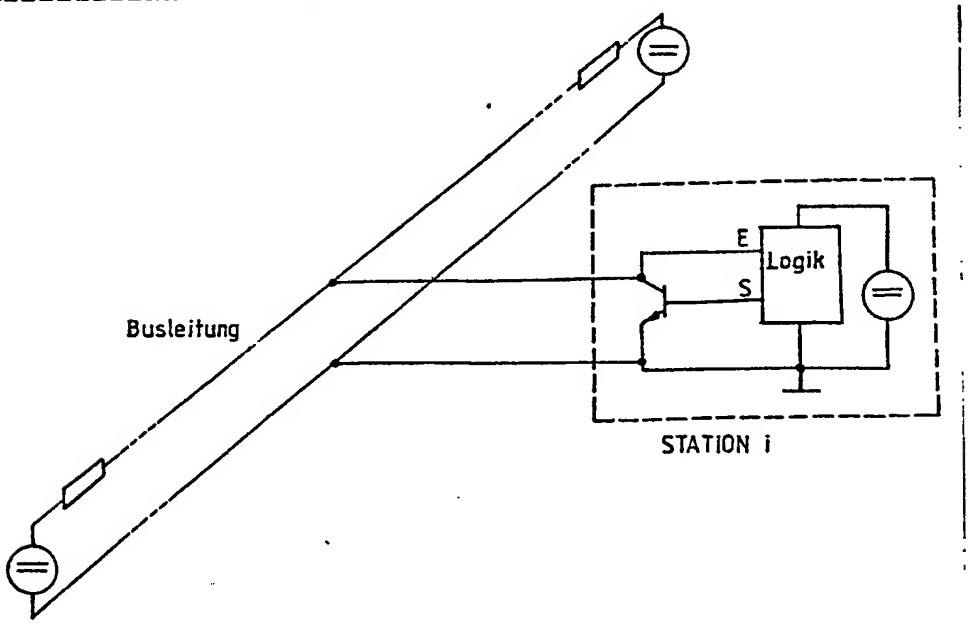
5. Zeichnungen

5.1. Ausfuehrungsbeispiel fuer lineare Busstruktur



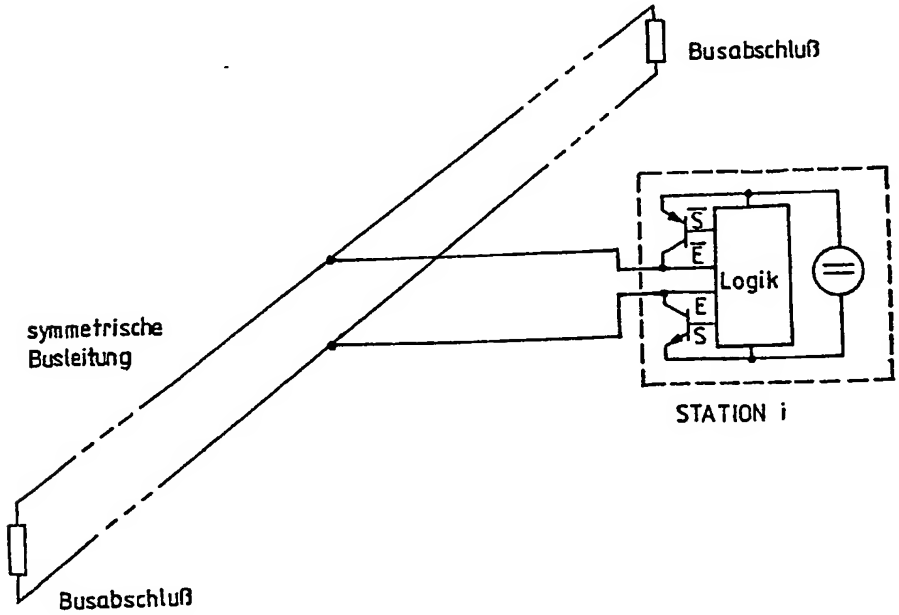
Figur 1

5.2. Beispiel fuer galvanisch gekoppelte, asymmetrische Ansteuerung der Busleitung



Figur 2

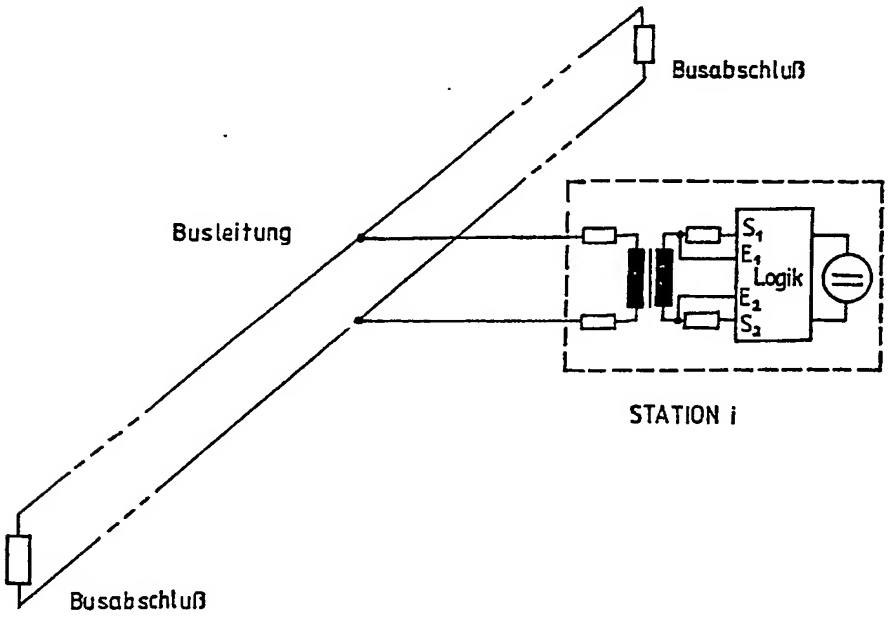
5.3. Beispiel fuer galvanisch gekoppelte, symmetrische Ansteuerung der Busleitung



Figur 3

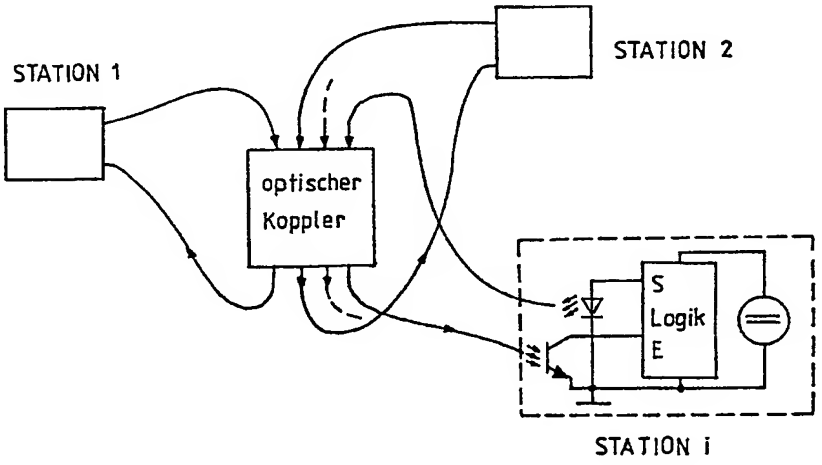
10.10.85
3506118

5.4. Beispiel fuer galvanisch getrennte, symmetrische Ansteuerung mit U bertrager



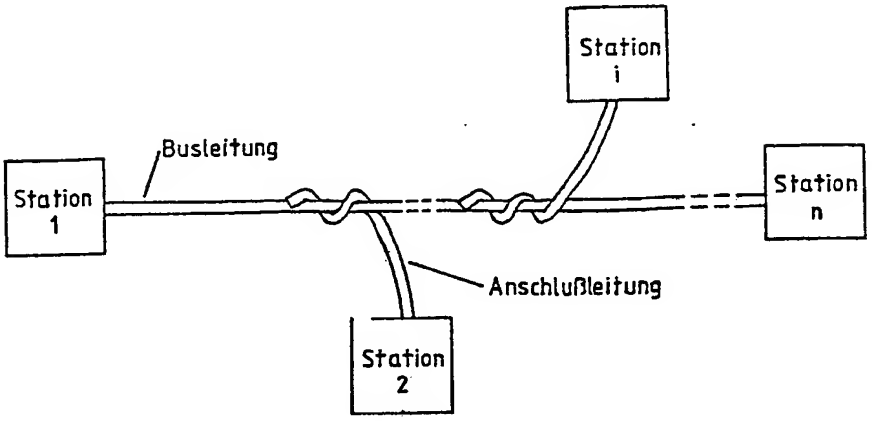
Figur 4

5.5. Ausfuehrungsbeispiel fuer Lichtleiter - Stern



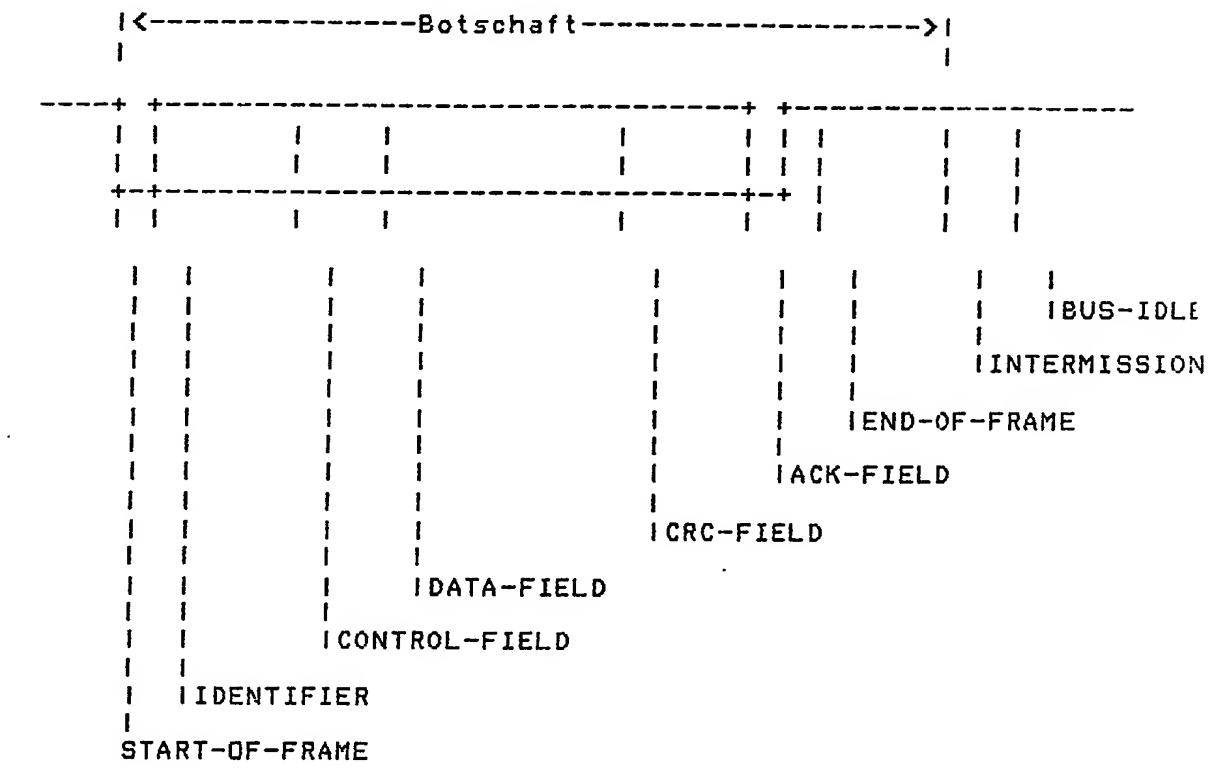
Figur 5

5.6. Ausfuehrungsbeispiel fuer Lichtleiter - Bus



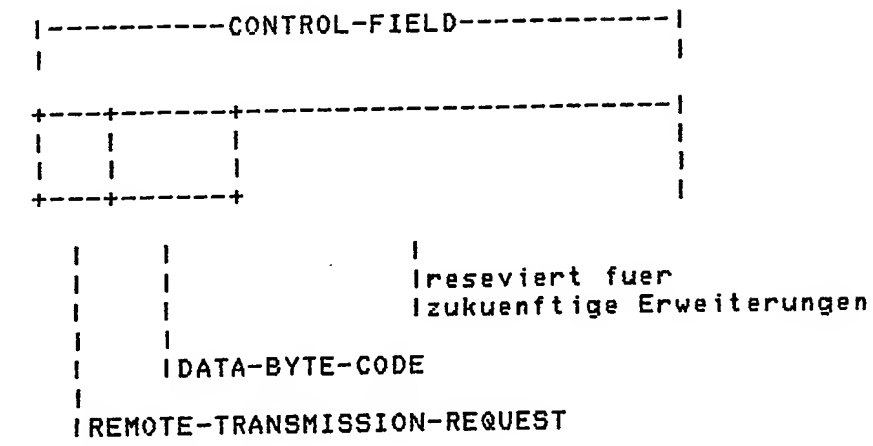
Figur 6

5.7. Aufbau einer Botschaft



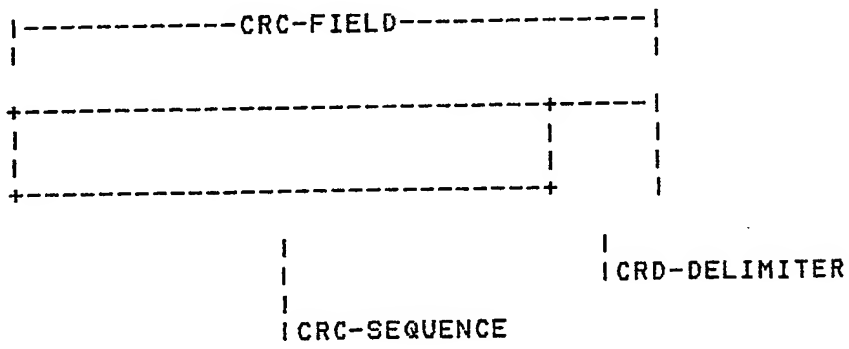
Figur 7

5.8. Aufbau CONTROL-FIELD



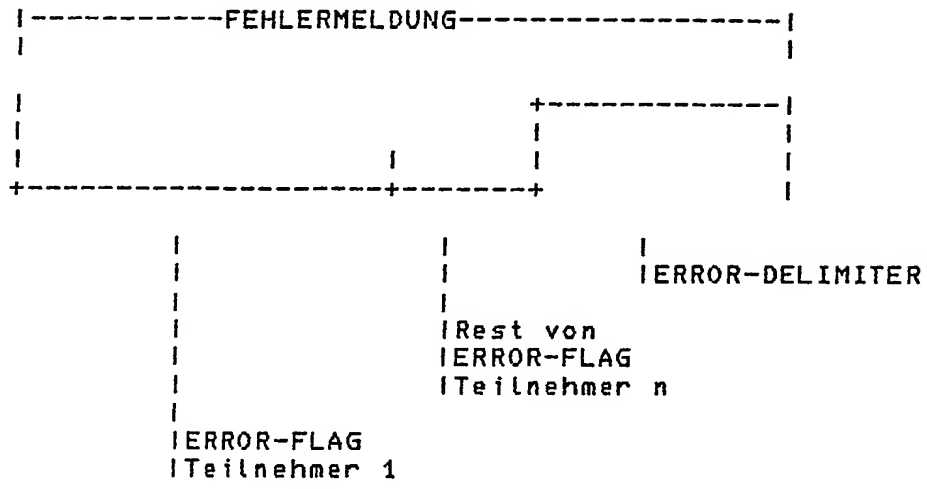
Figur 8

5.9. Aufbau CRC-FIELD



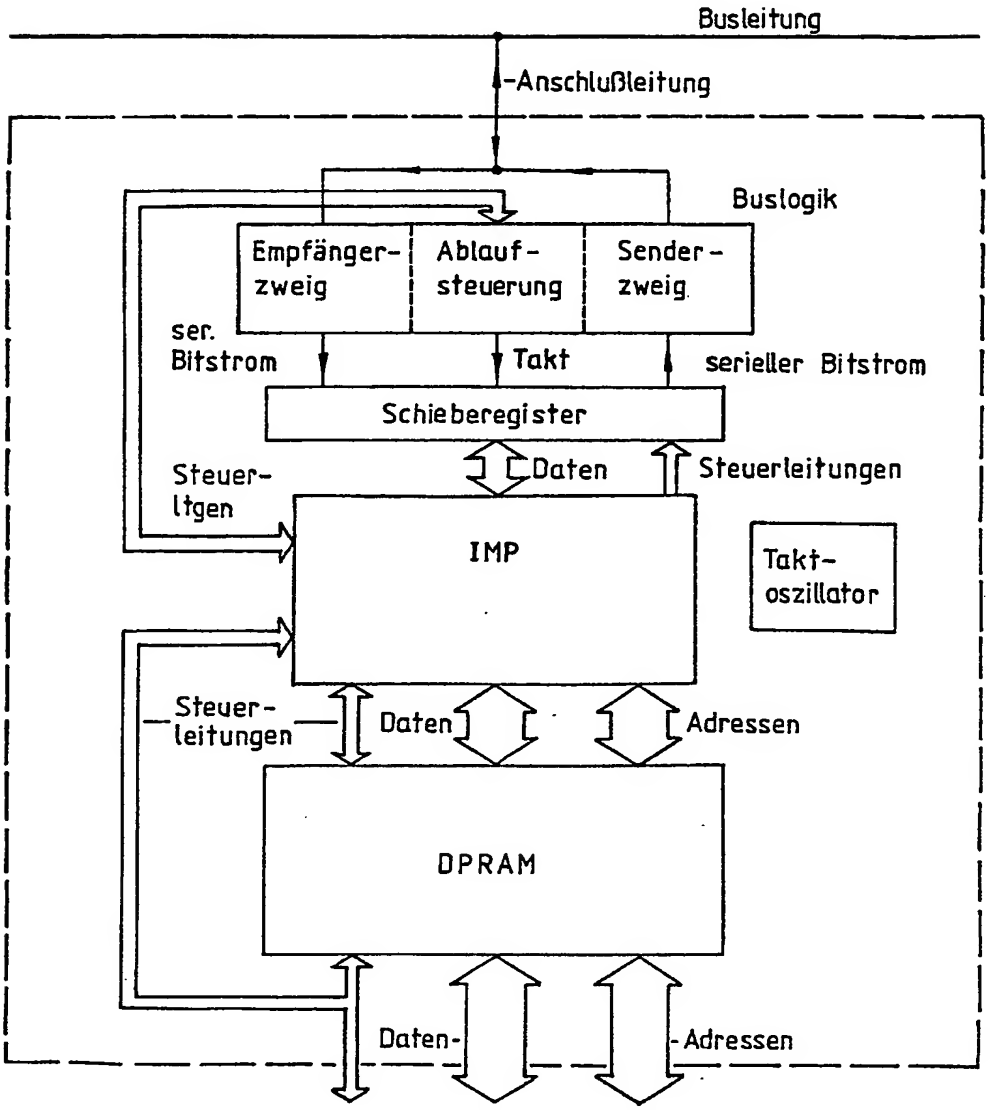
Figur 9

5.10. Aufbau einer Fehlermeldung



Figur 10

5.11. Blockschaltbild des seriellen Schnittstellen - Bausteins



Figur 11

Method for op rating a data processing system

Patent Number:	US5001642
Publication date:	1991-03-19
Inventor(s):	KIENCKE UWE (DE); DAIS SIEGFRIED (DE); KRAMPE WOLFGANG (DE); LITSCHEL MARTIN (DE); BOTZENHARDT WOLFGANG (DE)
Applicant(s):	BOSCH GMBH ROBERT (DE)
Requested Patent:	DE3506118
Application Number:	US19860831475 19860220
Priority Number (s):	DE19853506118 19850222
IPC Classification:	F02M51/00; G06F7/76
EC Classification:	F02D41/26D, F02P5/15B, G06F11/00B5, G06F11/07P4, G06F11/10, G06F13/374, H04L12/413B, H04L25/02G
Equivalents:	FR2578070, JP1989253C, JP2041107C, JP2545508B2, JP61195453, JP6236328, JP6236333, JP7021784B, JP7072883B

Abstract

A method is disclosed for the operation of a data processing system for motor vehicles including at least two computers and a line connecting the computers for the transmission of messages. This line permits a fast and reliable data transmission between the computers installed in the motor vehicle, taking into account the specific requirements of a controller-coupling in the motor vehicle. An embodiment is provided which describes in detail the interface between the individual computers and the line linking the computers, and with the aid of which a controller-coupling is realized in the vehicle.

DOCKET NO: WUP-IFT-808
SERIAL NO: _____
APPLICANT: Eric Pihet

LERNER AND GREENBERG P.A.
P.O. BOX 2480
HOLLYWOOD, FLORIDA 33022
TEL. (954) 925-1100